AFIT/GOA/ENS/95M-02

ANALYSIS AND MODELING OF AN AIRPORT DEPARTURE PROCESS

THESIS

Joseph E. Hebert, Major, USAF

AFIT/GOA/ENS/95M-02

DISCLAIMER STATEMENT

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U. S. Government.

AFIT/GOA/ENS/95M-02

ANALYSIS AND MODELING OF AN AIRPORT DEPARTURE PROCESS

THESIS

Presented to the Faculty of the Graduate School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Operations Research

Joseph E. Hebert, M. A.

Major, USAF

March 1995

Approved for public release; distribution unlimited

THESIS APPROVAL

STUDENT: Joseph E. Hebert CLASS: GOA 95M

THESIS TITLE: Analysis and Modeling of an Airport Departure Process

DEFENSE DATE: 28 Feb 95

COMMITTEE:	NAME/DEPARTMENT	SIGNATURE
Advisor	Lt Col Dennis C. Dietz/ENS	Den C Dietz
Reader	Dr Peter W. Hovey/ENC	Peter W. Hovey

PREFACE

The purpose of this study is to develop an accurate analytical model for the aircraft departure process at a major airport. The primary need was for a model which improves understanding of the performance of an airport departure queue.

The models developed provide an improved capability to determine an effective service rate for departing aircraft. In addition to providing a more accurate representation of the actual process, the models should improve a user's ability to predict the time and magnitude of the occurrence of significant patterns of delays. The models also provide a great deal of flexibility in the modeling of the airport departure operations by employing a modeling technique called the method of stages. This method enables the user to input more accurate probability distributions for the service time and still maintain the advantages of a Markovian model. The models also employ solution algorithms which improve solution times over the methods used previously.

There are several people whose support and guidance was crucial to the success of this thesis. First of all I would like to thank my thesis advisor, Lt Col Dennis Dietz, for his superior instruction and guidance. He kept me focused on the task at hand, encouraged me when things were going well, and urged me on when I needed it most. I also need to thank Dr. Peter Hovey for his support and insights. Finally, I thank my wife Marsha and my children, Jessica and Brandon, for their understanding. More importantly, I thank them for occassionally reminding me that I had a life other than working on this research.

Joseph E. Hebert

Table of Contents

List of Figures

Figure

Page

List of Tables

Table

Page

AFIT/GOA/ENS/95M-02

Abstract

This study develops improved analytical models to represent the non-homogeneous aircraft departure process at a major airport. Previous models have assumed that the system entry process (demands for service) and the system service process (take-offs) for the aircraft departure system are strictly Markovian. The models developed in this study expand on these Markovian models by employing the method of stages. This technique increases the model's ability to accurately represent the service time distribution, while maintaining the advantages of the Markovian model. In addition, the models in this study employ solution algorithms which are much more efficient than methods previously used. The models were developed using data collected at LaGuardia Airport in June 1994.

Three different models are developed and compared. All three assume a Markovian system entry process, but they all utilize different methods to represent the service (take-off) process. The first model assumes a Markovian service process. The use of this representation is shown to be reasonable and the results provide a good correlation to the actual system performance observed. The second model employs an Erlang distribution to represent the service time. This distribution supports the use of a Markovian model through the use of the method of stages. At the same time, the use of the Erlang distribution affords the model user the flexibility to better match the actual service time distribution in order to create the most accurate model. This model's results

also correlate well with the actual airport operations. Finally, a server absence model was developed. This model explicitly represents the periods of time when the runway (server) is unavailable to service the departing aircraft. Although this model showed promise of being the most accurate of the three, it also proved to be the most difficult and time consuming to employ.

The key feature of the aircraft departure process is its non-homogeneous nature. As a result, the most important aspect of the models developed is their ability to generate expected queue performance measures after finite amounts of time. These models should improve the user's ability to determine the effective service rate and to predict the occurrence of future patterns of delay.

ANALYSIS AND MODELING OF AN AIRPORT DEPARTURE PROCESS

1. Introduction

1.1 Background

The Federal Aviation Administration (FAA) has expressed a need to better understand the factors that can lead to aircraft taxi delays. In 1994, they requested research support from the Air Force Institute of Technology to study this area. In particular, the FAA wanted an accurate representation of the departure delay process in order to better understand how the process behaves and to identify the most significant causal factors. LaGuardia Airport was selected to provide the data for this study because it experiences significant departure delays and yet it does not have an excessive amount of traffic. The data was recorded from 1 to 7 June, 1994.

1.2 Initial Research

Initial analysis of the data revealed that the departure delay process was very time dependent. Since the rate at which aircraft left their departure gates varied greatly with the time of day, the occurrence of the most significant delays varied likewise. In addition, it was discovered during the initial research that several institutions have been developing methods for predicting aircraft take-off times for use by the Enhanced Traffic Management System (ETMS). Therefore, this research effort focused on developing an analytical model which describes and predicts the occurrence of significant patterns of aircraft departure delays. In addition to providing a more accurate means of airport

capacity estimation, this study should complement ongoing take-off time prediction efforts.

1.3 Definition of Terms

In order to improve the reader's understanding of the analysis to follow, the terms that will be used later in this report are now defined.

Departure queue -- The line consisting of aircraft waiting for their turn to take off.

Departure system -- The entire departure process being modeled. This includes all aircraft in the departure queue and any departing aircraft that has exclusive use of the departure runway environment. An aircraft enters the system when it joins the end of the departure queue or it is given immediate clearance to take off (in the absence of a queue). An aircraft departs the system when it has completed its take-off and cleared the runway environment sufficiently for another aircraft to be granted take-off clearance.

Pushback -- The point in time when an aircraft is pushed away from the departure gate so that it may commence taxi-out. This is also known as the gate departure time.

Demand for service -- The point in time when an aircraft is ready to be granted access to the runway. This does not imply that the runway is available for this aircraft to use. If other aircraft are already waiting for service, then the occurrence of a demand for service means that an aircraft has entered the end of the departure queue to wait its turn for take-off.

Taxi-out time -- The time interval between pushback and demand for service.

Roll-out time -- The time interval between pushback and the start of the aircraft take-off. This time includes taxi-out time and time spent waiting in the queue.

Aircraft departure -- An aircraft take-off.

Aircraft arrival -- An aircraft landing.

Airport departure capacity -- The maximum sustainable take-off rate at the airport for a given set of conditions.

Fractional departure capacity -- The effective maximum sustainable take-off rate for the aircraft from the eight major airlines which are represented in the data set.

1.4 Problem Statement

The aircraft departure process at major airports needs to be better understood. In addition, improvement is needed in the ability to predict aircraft departure delays under various conditions.

1.5 Objectives

The primary objective of this study is to develop an accurate analytical model of the aircraft departure process at LaGuardia airport. Related objectives involve demonstrating how this model improves understanding of the departure process and how the model might be useful for delay prediction.

1.6 Scope

The models developed are based on one week of data from LaGuardia airport. These models should improve the estimation of airport departure capacity experienced under various conditions. In addition, the models and methods used in this study should

facilitate the accurate prediction of aircraft departure delays. Finally, the models should aid in the identification of the most significant factors for airport departure delays. Although this research effort is based on the modeling of a single runway departure process, the modeling approach used should have applications for other situations and other airports.

1.7 Approach

Initial data analysis reveals that the aircraft departure delay patterns are clearly non-stationary. Therefore, it is evident that a useful model of this system requires transitory analysis. Since it is possible to estimate the transitory state probabilities for a Markovian model, primary attention is focused on developing an accurate Markovian model of the system.

All of the models developed in this study are single server queuing models with Markovian entry (demand for service) processes. LaGuardia has two intersecting runways. Normal operating configurations assign one runway as the primary departure runway. Therefore, the models developed all use a single server. Determination of the demand for service process is relatively straight forward. It is determined to be closely related to the pushback process. When the pushback data is analyzed in blocks of time for which the rate appears homogeneous, the process is discovered to fit a Poisson distribution, which supports a Markovian model.

The determination of the service (take-off) rate is more difficult. It is assumed that there was always a queue present during the periods of time when a significant pattern of aircraft departure delays were observed. When the time between aircraft take-offs is

analyzed during these conditions, it is found that the empirical distribution can be reasonably modeled with an exponential distribution. However, it is also apparent that these times can be better represented by an Erlang distribution or a probabilistic mixture of the Erlang distribution and a convolution of Erlang distributions. These service time representations are the foundation of the two primary models used in this study.

The service process is modeled in stages, which allows the above representations for the service time distributions to be used in the Markovian models. The transitory state probabilities for these Markovian models are approximated and then used to estimate queue lengths and waiting times. The analysis results are based on the comparison of the model inputs to the model outputs and the actual observed queue conditions. The solution methods used are based on an approximation technique known as uniformization (16:286)

1.8 Summary of Results

When the models developed are applied to the departure delays observed at LGA, it becomes evident that the effective service rate for the aircraft from the eight major airlines represented in the data set was different than reported. Most of the time, a single value of this fractional departure capacity is fairly accurate in reproducing the pattern of departure delays observed. It will be shown that these models help identify the actual departure capacity observed under a certain set of conditions.

When the fractional departure capacity is accurately estimated for the conditions, it is possible to predict the pattern of future delays. The estimate may be determined from a historical application of these models, or to the recent system conditions experienced at the airport. In other words, the closest estimate for the departure capacity from an hour

just ended might be used as the departure capacity for the next hour. In addition, the current observed departure delays could be used to define the current state of the system. This state could then be used as the initial condition in the model run for the next time period. It will be shown how the models developed can aid in the determination of the airport's effective capacity and how these models might be useful for the prediction of actual queues and delays.

1.9 Thesis Outline

Chapter 2 provides a literature review for applicable works in the areas of airport delay analysis and queuing theory solution techniques. Chapter 3 provides a detailed analysis of the LaGuardia data set used as the basis for this research effort. It includes initial analysis conclusions, and appropriate probability distribution models for the processes observed. Chapter 4 presents the development of the three models used in this study: the M(t)/M/1 queue model, the M(t)/Ek/1 queue model and the M(t)/Ek/1 queue model with random server absences. Chapter 5 goes into detail on the estimation of the queue performance measures for each model. The discussion starts with an explanation of how the rate matrix is created and how it is used to estimate the transition matrix. The development of sequential state probability vectors for the aircraft departure system is presented next. Finally, the queue performance measures calculations are explained for each of the three models. In Chapter 6, the results are shown by demonstrating how the models perform for the LaGuardia airport data. Chapter 7 concludes with suggested applications and recommendations for future research.

2. *Literature Review*

This chapter contains a summary of selected literature which is relevant to this thesis. The works are separated into three groups. The first group consists of articles which address aircraft delay analyses. The second provides a review of past efforts at general analytical modeling. The third group consists of articles which address other airport operational issues such as airport capacity estimation, and take-off time prediction.

2.1 Aircraft Delay Analyses

Herbert Galliher and R. Clyde Wheeler (1958) offer one of the earliest attempts at using numerical solution methods to help describe the transient operation of an airport landing queue. The application they provide assumes that the entry into the queuing system is a Poisson process, while the servicing (landing) process has a fixed length time between events. Their primary emphasis is on the transitory effect of time-variant aircraft arrivals to the airport and how they affect the resulting queue. They do not address the effect of various service process representations.

Bernard O. Koopman (1972) performs an analysis of the effects of time-varying demand on the queue of airborne aircraft awaiting landing clearance. He points out that most previous efforts in the area addressed the problem as a stationary process. In his study, he uses two different types of service time representations. The first representation he uses is a service time of fixed length. The second is a service time that is exponentially distributed.

Koopman draws two main conclusions from his study. The first is that the results of the single waiting line queue are "highly sensitive to servicing rate," but are rather "insensitive to the statistical assumptions concerning the law of service." The second conclusion is that a double queue (one for take-offs and one for landings) can be analytically reduced to a fairly accurate single queue. (10:1105)

Emily Roth (1979) performs a more detailed analysis of the time variant behavior of an airport queuing system. Roth's model is more advanced than Koopman's because she attempts to take into account variation in service time due to different spacing requirements for different combinations of aircraft. She also models the system with two queues, one for arrivals and one for departures. She argues that her "extended," two-queue model is a more accurate representation of the actual delay process.

Roth uses her model to analyze the effect of three different priority schemes: landing priority, departure priority, and alternating priority. She points out, however, that one of the major limitations of her model is the time required to solve a typical problem. As an example, she points out that the execution of the extended model for 18 hours of data, and a maximum queue length of 13 aircraft, required 12.7 minutes of CPU time. (18:55)

Robert C. Rue (1979) uses a Semi-Markov decision process to show the advantages of a using the social optimum to control aircraft arrival access to an airport. Without social optimum control, each individual customer decides to enter the queue based on maximizing his own gain. With social optimum control, the entrance decision is based on maximizing the benefit for all customers. Rewards and costs are assigned to

each individual aircraft that enters the queue. These quantitative measures are then used to determine the optimum priority policy for the various classes of aircraft. (19:2)

2.2 Analytical Modeling

Donald Gross and Douglass Miller (1984) present a method for achieving a transient solution to discrete state space, continuous time Markov processes. They assert that much of the analysis that has been performed on Markov processes has been limited to the analysis of the equilibrium condition. The authors point out that many processes require transitory analysis in order to properly represent the system. Such analysis may be called for when systems encounter time-varying system parameters which affect the system's performance. Transitory analysis may also be warranted if the system proceeds to equilibrium so slowly that steady-state analysis does not adequately describe the system's performance. (7:343-344) The modeling and solution technique the authors present is based on a solution method known as randomization. This method has been called uniformization by other authors (17:174-176).

The solution methodology Gross and Miller offer generates a numerical solution to the set of differential equations which characterize a Markov process in transition. Their method requires the computation of many individual terms of an infinite series in order to achieve a reasonable approximation. These intermediate computations can require a large number of time consuming matrix multiplications. The authors address this problem by truncating the series when the probability value for a term is less than a predetermined tolerance level.

The authors point out that solution efficiency may be improved if several steps are taken. First of all, the problem may be simplified through the linearization of the state space. If the most accurate model of a system is multi-dimensional, it may be possible to simplify its solution by ordering the states "into a one-dimensional state space" (7:350). Solution efficiency may also be improved through the implementation of matrix multiplication algorithms which exploit the sparse nature of the matrices involved.

J. Medhi (1991) describes a special class of stochastic models which are called queueing systems with vacations. In this type of queueing model, the server is periodically unavailable to service customers. Although the general concept of a server vacation (or absence) is of interest to this thesis effort, existing models are not useful for several reasons. First of all, the models assume that server vacations only occur when the queue is empty. Secondly, only the equilibrium condition of a modeled system is addressed. (12:399)

2.3 General Air Traffic Analyses

Amedeo Odoni (1987) provides a general discussion of airport capacity estimation and aircraft delay optimization. One of his primary conclusions is that "optimization tends to favor large aircraft (biases) and long flights" (13:283). He further states that this systematic bias is one of the most commonly encountered problems with attempts to apply optimization to airport operations. Odoni also identifies some of the primary variables for airport capacity estimation. These include the weather, the operating runway configuration, ATC separation requirements and procedures, traffic mix, runway geometry and human factors. (13:274)

Robert Shumsky (1993) provides an initial analysis of take-off time data at Logan Airport, collected in March, 1991. This analysis is in support of research he is currently doing for the FAA to improve the agency's ability to predict of take-off times. His research effort addresses several of the causes for delays in take-off time. These include departure delays caused by late arrival to the airport from a previous flight and ground delays due to departure demand exceeding capacity. His discussion identifies some of the limitations with the prediction method now in use. In his report, he demonstrates the importance of runway configuration and capacity information to any type of meaningful analysis of the system.

Eugene Gilbo (1993) provides an empirical method of estimating an airport's operating capacity. Gilbo's approach involves analyzing the observed number of departures and arrivals over a fixed time period. This method requires that the data be collected when the airport is operating at near peak capacity. Gilbo assumes this to be the case whenever the data indicates the existence of significant delays. (6:145)

Gilbo explains that peak operating capacity may periodically surge beyond rates which are sustainable. Therefore, his estimates are determined after rejecting extreme outlier observations. He argues that doing so helps to improve the robustness of his technique. He then extends his analysis to show how the resulting capacity estimates may be used to improve the allocation of scheduled departure and arrival times and to better satisfy traffic demand.(6:153)

2.4 Summary

The aircraft delay problem has received considerable attention over the last several decades. Studies have been heavily focused on the overall air terminal delay situation and the determination of policies which will minimize delays. The efforts in this area have included simulations and analytical models to characterize and predict these delays. The literature considered in this review has been limited to those studies which analytically model the queue and the delay process. Most of the models presented employ Markov or Semi-Markov processes to accurately represent the air traffic situation. Although none of the literature reviewed focuses solely on the ground delay problem, the insights obtained on the nature of airborne delays may be directly applicable to a ground delay model representation.

3. Data Analysis

3.1 Data Description

The data used for this study comes from LaGuardia Airport (LGA). LaGuardia was chosen because it has a significant amount of traffic and periodically experiences delays in the departure queue. LaGuardia has two runways. During normal flying operations, one runway is designated as the departure runway and the other as the arrival runway. This feature helps reduce the complexity of the problem by allowing the use of a single server queue model. Although departures are generally given exclusive use of one runway, the runways intersect, hence the arrival process has some impact on the departure process and must be accounted for in the final model.

The primary data used for this study was provided by the MITRE corporation. The data set uses inputs from the Aircraft Radio Incorporated (ARINC) Communications Addressing and Reporting System (ACARS) and the Airport Service Quality Performance system (ASQP). It includes 3885 records of arrival and departure data for the eight major airlines that flew into, or out of, LGA from 1 to 7 June, 1994. Each record includes 12 fields of primary data and 27 fields of derived data. The primary fields are: airline, flight number, departure airport, arrival airport, date, Official Airline Guide (OAG) departure and arrival times, actual departure and arrival times, departure message time, arrival message time, and aircraft type. The derived fields represent values that are computed from the above primary data fields. The derived data fields include: wheels-off and wheels-on times, elapsed flight time, total delays, and specific delays for pushback, taxi-

out, and take-off. The primary data set is recorded in Greenwich Mean Time (GMT) minute format. This means that the time recorded for an event occurrence was the number of minutes that had elapsed since midnight in Greenwich England.

In order to attempt to correlate departure delays to factors deemed important, copies of the daily airport records, hourly weather observations and airport operating configuration (primary departure and arrival runways) are included for the seven day period of the study.

3.2 Initial Analysis

Since the primary focus of this study was the departure process and the resultant departure delays, most of the analysis is performed using the 233 to 294 daily departure records. Due to recording problems on 1 and 2 June, much of the data from those days is missing. However, sufficient data is available on the remaining days to create useful models and generate insightful conclusions. The most interesting two days in this data set are 6 and 7 June. These are the only days that experience any significant weather phenomena and they are also the days that experienced the most significant delays.

The first analysis performed is to study the plot of actual delays experienced versus the time of day. The data manipulation and plots are accomplished using Microsoft Excel. The data file is first separated into arrival and departure data. Next, the departure data is separated by day of the week. Finally, the actual roll-out times are plotted against the time of the day that each flight actually left their gate (pushback time).

Since all of the time of day data are recorded in GMT, these times are converted to local hour/decimal hour format to facilitate analysis. The conversion factor to convert

GMT to local time for LGA in June is 4 hours (3:B-282). Therefore, local time is computed by subtracting 240 minutes from the recorded GMT time to obtain a local time in minute format. This number is then divided by 60 to generate the decimal hour result.

In order to produce the plots described above, the roll-out times had to be computed since the data set does not include this data field. The data set does, however, include a taxi-out delay data field which is computed as the difference between the actual gate departure time and the actual wheels-off time minus a nominal 15 minute taxi-out time. The wheels-off time, which is itself a derived data field, is computed by subtracting the departure gap time from the departure message time. The departure message is generated when the ground controller first identifies the airborne aircraft on radar. Although the data set includes a taxi-out delay data field, the actual roll-out times are plotted in order to avoid the need to plot negative taxi delay values. The plot for Monday, 6 June 94 is shown in Figure 3.1. Plots for all seven days may be found in Appendix 1.

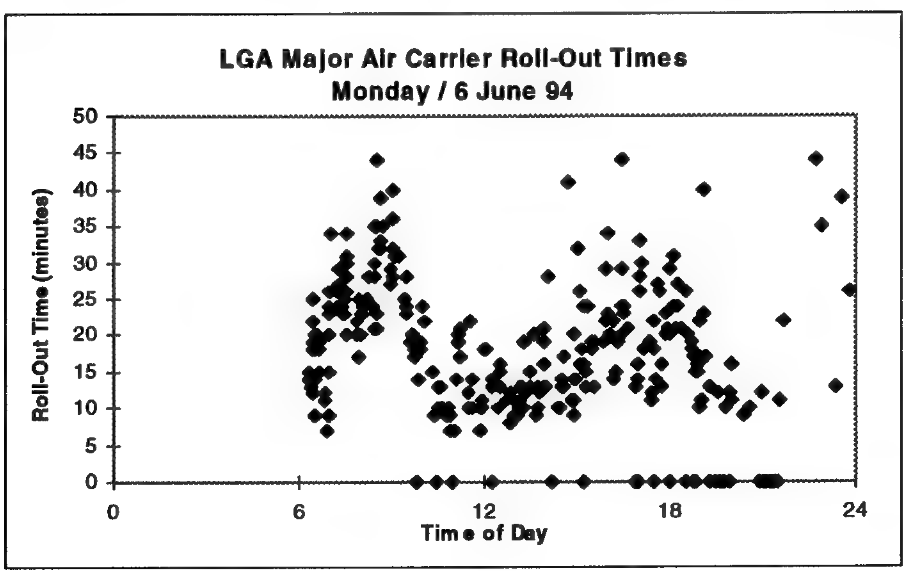


Figure 3 .1 Roll-Out Times -- 6 June

Due to problems with the data collection process and the merging of the two data sets mentioned, some of the records are missing the necessary fields to compute the roll-out time. These records are included in the plot to show where the missing data exists to provide better understanding of the data. These records are plotted as zero roll-out times. If a record's actual gate departure time is missing, these records are plotted at the OAG scheduled gate departure time.

The plots of actual roll-out times demonstrate that significant nonrandom delays occur on several days during certain times of the day. The most significant delays occur on Monday, 6 June, while the least significant delays occur on Saturday and Sunday, 4 and 5 June. The plots of data indicate the existence of peak periods in the departure process. Since 6 June has the most significant taxi delay pattern, it is the data set used to initially develop the model.

It is important to note the existence of missing data. The occurrence of missing records in the data for 6 June, represented by the zero roll-out times, shows the number is not great. Out of a total of 287 departure records for this day, 26 are missing, or 9 percent. For the seven days of data available, the amount of missing data varies from 3 percent on Saturday, 4 June, to 55 percent on Wednesday, 1 June. As mentioned previously, most of the missing data occurs at the beginning of recording period due to initial problems with the data collection process.

3.3 Demand Process.

The next aspect of the data to be analyzed is the demand for service process. Since the actual times that departing aircraft demand service are not recorded, the

pushback (gate departure) times are analyzed instead. The number of pushbacks per hour on 6 June is plotted. These plots show the existence of peak periods from 6:00 to 10:00 AM and again from 3:00 to 7:00 PM. When the time between these events are plotted, the data visually supports an exponential distribution fit. A plot of the times between pushbacks for the 6 June AM peak period demonstrates this observation.

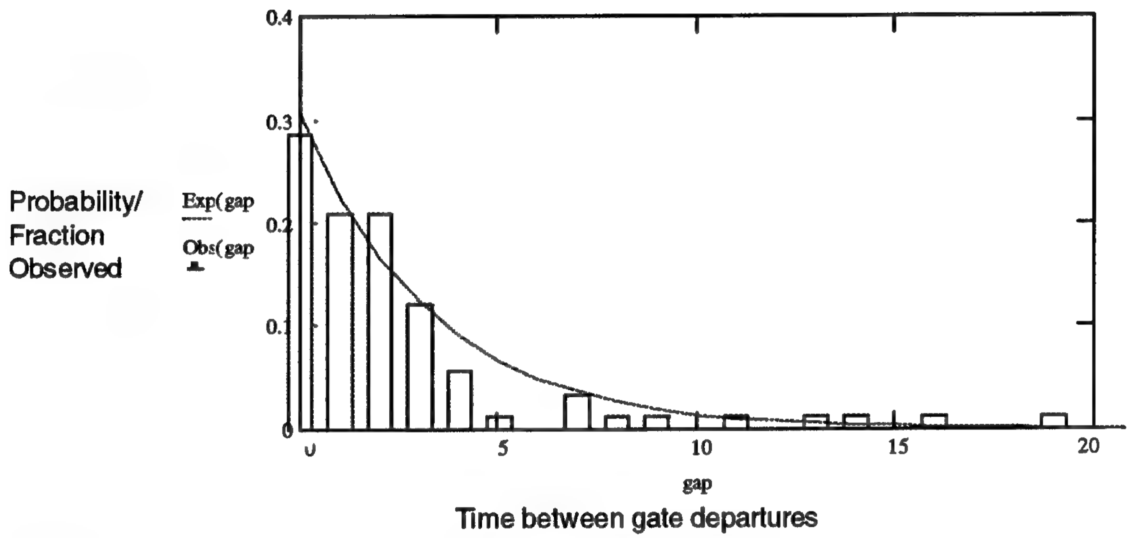


Figure 3.2 Time Between Pushbacks (Gate Departures) -- 6 June, 06:30 - 10:00

If the time between pushbacks is exponential, then this process is Poisson. Since the number of pushbacks per hour varies by the time of day, this process must be described as a non-homogeneous process.

The distribution of the time between pushbacks is analyzed in one hour blocks. The chi-square and the Kolmogorov-Smirnov (KS) goodness-of-fit tests are performed to see whether the data fails to "fit" the exponential distribution. Using a 5 percent confidence level, the chi-square goodness-of-fit test fails to reject the hypothesis that the data fits the exponential distribution for all of the hours between 6:00 AM and 8:00 PM on

6 June and the peak periods from 6 and 7 June. The KS test also fails to reject the same hypothesis for all of the hourly observations from 6 June. Therefore, the pushback process can be reasonably represented as a non-homogeneous Poisson process. The chi-square and KS goodness-of-fit test results are provided in Appendix 2. The summary of the KS goodness-of-fit test results are shown in the table below.

Kolmogorov-Smirnov goodness-of-fit test results
time between gate departures (pushbacks)
versus the exponetial distribution
Hourly Data from 6 June

Hour of the day			KS Statistic	Degrees of Freedom	Confidence Level (p - value)
6:00	to	7:00	0.208	24	greater than 0.1
7:00	to	8:00	0.206	22	greater than 0.1
8:00	to	9:00	0.229	21	greater than 0.1
9:00	to	10:00	0.201	22	greater than 0.1
10:00	to	11:00	0.345	14	greater than 0.05
11:00	to	12:00	0.206	17	greater than 0.1
12:00	to	13:00	0.181	14	greater than 0.1
13:00	to	14:00	0.156	17	greater than 0.1
14:00	to	15:00	0.267	15	greater than 0.1
15:00	to	16:00	0.189	18	greater than 0.1
16:00	to	17:00	0.229	16	greater than 0.1
17:00	to	18:00	0.139	18	greater than 0.1
18:00	to	19:00	0.229	22	greater than 0.1
19:00	to	20:00	0.133	12	greater than 0.1

Table 3.1 KS Goodness-of-Fit Test Results, Time Between Pushbacks

Now that it is concluded that the pushback process can be represented as a non-homogeneous Poisson process, it is necessary to relate this process to the actual demand process or the entry process to the departure queue itself. If it can be assumed that the vast majority of departure delays occur at LGA due to delays in obtaining clearance to take off, then most delays encountered from the gate to the end of the runway should be minor in comparison. Therefore, it is assumed that the taxi-out process from the gate to the

departure queue can be modeled as a multi-server queue that can handle as many aircraft as attempt to taxi out. If this is the case, then the taxi-out process would resemble an infinite server system. This, along with the earlier assumption that the pushback process is a Poisson process, allows the conclusion that the output process from this system is also a non-homogeneous Poisson process regardless of the distribution of the service time (taxi-out time). Finally, since the output of the taxi process is the input to the departure queue, it can be concluded that the entry process to the departure queue also follows a Poisson process. This, along with the previous assumption of a single server system for the departure process, are the key assumptions for the analytical modeling that follows.

In order to obtain an estimate for the taxi time, it is assumed that a queue did not exist when the roll-out times appeared to be stable and when the vast majority of these times were 20 minutes or less. The taxi times from 10:30 AM to 2:30 PM on 6 June meet these conditions. When these times were plotted, taxi-out time appears to be normally distributed with a mean of 13 minutes. Although this is not a very rigorous estimate, it should be sufficient for the purposes of this study.

3.4 Service Process.

The next aspect of the data to be analyzed, and the hardest one to characterize, is the actual service process (rate at which take-offs occur). This rate is the key to the accurate modeling of the departure process. It is also an area that has received a great deal of study in the past. An airport's capacity is a much sought after piece of information. The estimated capacity is a function of the operating runway configuration and the weather. For example, LaGuardia's best runway configuration is to conduct

arrivals on runway 22 and departures on runway 13. When the weather allows Visual Flight Rules (VFR), LaGuardia can handle 32 to 36 arrivals per hour, or roughly one every 1:42 minutes. When LaGuardia is conducting arrivals on runway 13 via the Instrument Landing system (ILS) and departures on runway 13, the arrival rate capability is only 18 to 22 per hour. The rate of aircraft departures is usually comparable to the rate of arrivals. (4:1-2)

The problem with estimating the departure process from the take-off times in the data set is the fact that the time between take-offs is not the "service time" in the queuing model. There are several reasons for this. The first is the fact that there may be times during the day when the queue empties out and there are no aircraft waiting to take off. In this case, the time between take-offs includes some idle time for the runway (as far as the departure process is concerned). The second problem with estimating the service time is the fact that there are several outside influences which can increase the time between aircraft departures. For LaGuardia this includes safe clearance requirements for the aircraft landing on the intersecting runway and in-flight safe separation requirements for aircraft flying into, and out of, airports in the New York Terminal Control Area (TCA). These airports include JFK and Newark International Airports. One final factor to be mentioned here is that the data set does not include the same level of detail for General Aviation Aircraft (GAA), although these aircraft are part of the same departure queue being studied. The representations of service time used in this study attempts to account for these factors.

The time between takeoffs is analyzed for the peak periods of 6 and 7 June. Since all aircraft taking off during these times experience roll-out times greater than average, it is assumed that the runway was never idle due to a lack of aircraft ready to take off. In other words, it is assumed that there was always a queue present during these peak periods. A histogram is generated for the time between take-offs for these peak periods and visually compared to exponential and Erlang-k distributions. The Erlang-k distribution represents the sum of k independent, identically distributed exponential random variables, where k is an integer. The graphs for the AM and PM peak periods for 6 and 7 June are provided in Appendix 1. Figure 3.3 below shows the data for the AM peak period on 6 June.

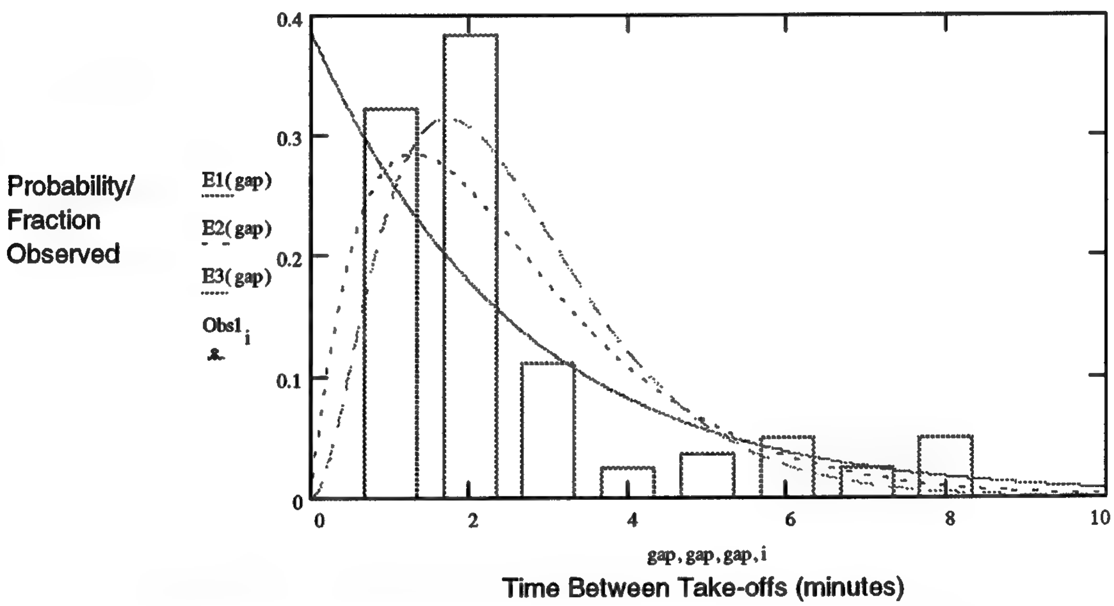


Figure 3.3 Time Between Take-offs -- 6 June, 06:30 - 10:00

The E1, E2 and E3 graphs represent the exponential, Erlang-2 and Erlang-3 distributions respectively. All three are computed with a mean equal to the mean of the

empirical data. The histogram represents the fraction of observed take-off gaps that were "i" minutes apart.

These graphs indicate that the data may fit either an Erlang-k or exponential distribution, with the Erlang distributions appearing to achieve the better fit. When the first two moments of the distributions are matched to the first two moments of the observed data, the Erlang-2 distribution fits the closest. However, when the chi-square and KS goodness-of-fit tests are performed, neither the Erlang-2 nor the exponential distribution are rejected. The goodness-of-fit test results are provided in Appendix 2. Since the KS goodness-of-fit test is more powerful than the chi-square test, the summary of the KS test results are listed in Table 3.2.

Kolmogorov-Smirnov goodness-of-fit test results
time between take-offs
peak periods on 6 and 7 June

Date	Time of Peak			KS test Statistic	Degrees of Freedom	Confidence Level (p - value)
6-Jun	6:30	to	9:00	0.103	58	greater than 0.1
6-Jun	15:00	to	18:00	0.088	51	greater than 0.1
7-Jun	6:30	to	9:00	0.139	63	greater than 0.1
7-Jun	16:00	to	18:00	0.086	35	greater than 0.1

Table 3.2 KS Goodness-of-Fit Test Results, Time Between Take-offs

Intuitively, these distributions make some sense. If the airport departure capacity were 40 per hour, we could expect that departing aircraft could take off once every 1:40. However, the data indicates that the time between takeoffs is often greater than this and sometimes it is a good deal more. Whether they are due to GAA departures, arriving aircraft, or other outside influences, the resultant delays appear to follow one of these

distributions. Therefore, it is assumed that these random gaps may be adequately modeled with the exponential and/or the Erlang distributions.

It is possible that a better representation for the service time distribution may be obtained as the probabilistic mixture of two separate conditional distributions. The first conditional distribution would represent the time between take-offs given there were no outside delay factors (e.g. GAA departures, arrivals, etc.) In this case, the distribution represents the time between take-offs that could be attributed solely to departure aircraft spacing requirements. The second conditional distribution represents the time between take-offs, given that there is a random occurrence of one of the outside influences mentioned previously, such as landing aircraft or GAA departures. In this case, the time includes the amount of time that the server (the runway) is not available for take-offs for the aircraft in the queue, plus the amount of time for the aircraft to take-off once the server became available again. The random additional delays will be called server absences, because the server (runway) can be considered not available for the aircraft in the queue.

One of the significant features of the graph in Figure 3.3 is the fact that there is an accumulation of probability mass in the tail of the graph that is not characteristic of the exponential nor the Erlang distribution. In order to use the absence model described above to model this observation period, those times of three minutes or less are assumed to be service times occurring under the first condition (no server absence occurred). Those times that are greater than three minutes are assumed to be service times under the second condition (a server absence occurred). In the first case, matching the first two

moments yields an Erlang-6 distribution. Figure 3.4 demonstrates the fit of this distribution to the service times that were three minutes or less.

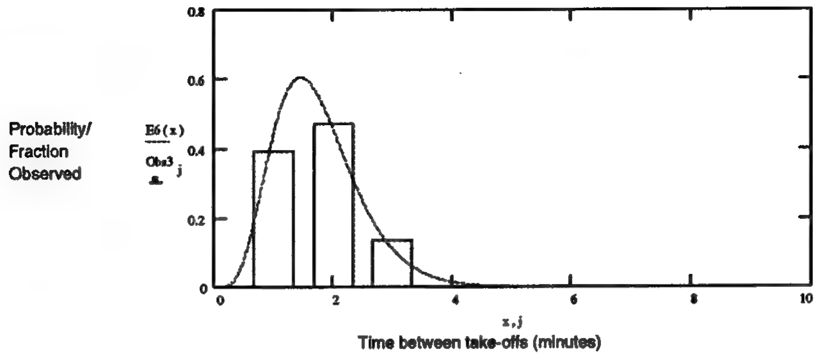


Figure 3.4 Time Between Take-offs (< 4 Minutes)

The next challenge is to fit the distribution for the service time, given an absence occurs. In this case, the distribution will be the convolution of the absence time and the service time once the absence is over. In order to estimate the distribution parameters for the absence time, a nominal two minutes of service time is subtracted from those times between take-off that were greater than three minutes. The Erlang-9 distribution is chosen by matching the first two moments of this distribution to the data.

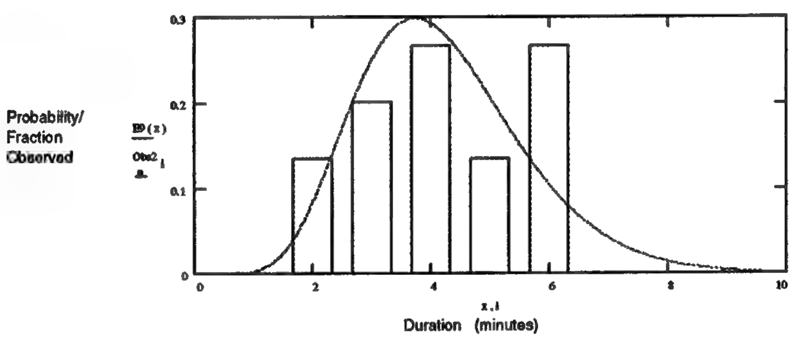


Figure 3.5 Absence Durations

The final distribution for the service time is the probabilistic mixture of the basic service time and the convolution of the basic service time and the absence time. Due to

the complexity of the analytical representation for the convolution of an Erlang-6 and an Erlang-9 distribution, Monte Carlo simulation is used to demonstrate the fit of the theoretical distribution to the data. This simulation is performed using 500 observations.

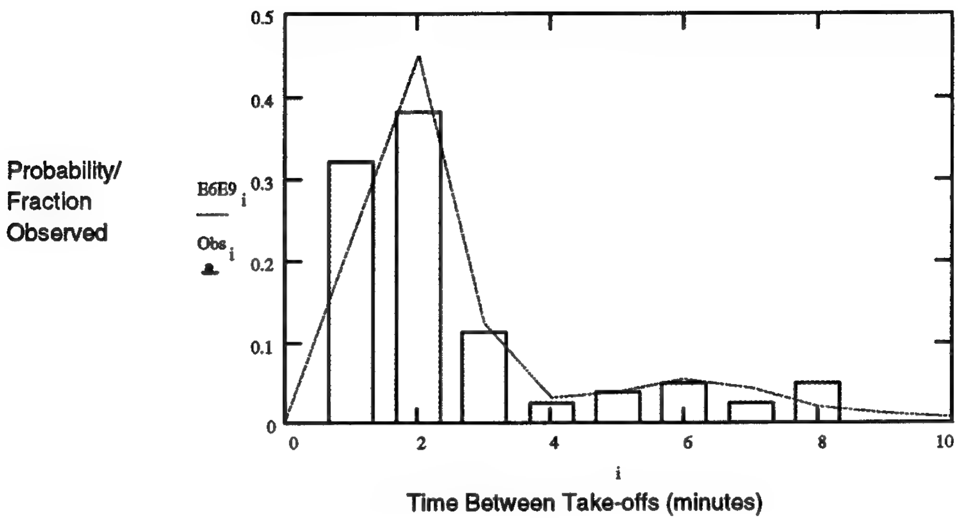


Figure 3.6 Service Time Representation -- Monte Carlo Results

This graph indicates that this form of a service time distribution might have merit.

3.5 Absence Occurrences

The final data analysis performed at this level is to determine if the absence occurrence process could also be modeled as Poisson. As was assumed previously, an absence is considered to have occurred when the time between take-offs is greater than 3 minutes during a peak period. The resulting times are tested for fit with the exponential distribution using the KS goodness-of-fit test. None of the data for four peak periods are rejected for this fit. The four periods are the AM and PM peak periods on 6 and 7 June. The KS goodness-of-fit test results are in Appendix 2. The summary of the test results is shown in Table 3.3.

Kolmogorov-Smirnov goodness-of-fit test results
time between absence occurances
peak periods on 6 and 7 June

Date	Time of Peak			KS test Statistic	Degrees of Freedom	Confidence Level (p - value)
6-Jun	6:30	to	9:00	0.241	13	greater than 0.1
6-Jun	15:00	to	18:00	0.102	13	greater than 0.1
7-Jun	6:30	to	9:00	0.225	15	greater than 0.1
7-Jun	16:00	to	18:00	0.16	15	greater than 0.1

Table 3.3 KS Goodness-of-Fit Test Results, Time Between Absences

As a result of these tests, the absence occurrence process is assumed to be adequately represented by a non-homogenous Poisson process.

3.6 Summary

In order to determine the feasibility of analytically modeling the departure process of the LaGuardia departure queue, the available data set had to be analyzed in detail. The first analysis accomplished was to determine the hourly pattern of departure delays. This was accomplished with a scatter plot of the roll-out times versus the hour of the day that each flight accomplished pushback. These plots clearly demonstrated the existence of peak periods for delays. In addition, they demonstrated that the delay process varied significantly from one day to the next. When the delay patterns did occur, they coincided with the peak periods for aircraft pushbacks.

The next data analysis performed was to determine an appropriate probability distribution for the pushback process. It was determined that this process closely resembles a non-homogeneous Poisson process. Both the chi-square and the KS goodness-of-fit tests were performed and both failed to reject the exponential distribution for the time between pushbacks in each hourly time interval. This process was then

related to the entry process for the departure queue with the conclusion that this process could also be considered a non-homogeneous Poisson process.

The time between aircraft departures during peak periods was used to estimate the service time for a departure queue. It was assumed that a queue was always present whenever the mean roll-out time for a period was greater than 15 minutes. Three different models were used to characterize these times. The first model assumed an exponential distribution for these times. The second model assumed an Erlang-2 distribution for service times. Neither the exponential nor the Erlang-2 distribution were rejected by the chi-square and the KS goodness-of-fit tests.

The third model for the service time distribution assumed that the times were actually the mixture of a service time and a server absence time. The distribution was estimated using an Erlang-6 distribution for the service time and an Erlang-9 distribution for the absence time. The total distribution was estimated using Monte Carlo simulation and visually verified as a possible fit for the data. In addition, the time between server absences was tested for exponential fit using the KS goodness-of-fit test. The fit of the absence process to a Poisson process was not rejected by this test.

Since it has been determined that the entry process to the departure queue adequately fits a Poisson process, and the service process can be adequately represented by exponential, Erlang, or some combination of Erlang distributions, it is now possible to develop models to describe the departure process using a Markovian state space. Chapter 4 will describe how these models are defined.

4. Markovian Modeling

4.1 General

One of the primary advantages of a Markovian model is that its transitory state probabilities may be readily estimated. A key feature of the LGA aircraft departure process is its obviously transitory nature. The rate at which aircraft arrive at the end of the runway to take off varies significantly with the time of day. In fact, it has become readily apparent that this variable rate, and the airport's occasional inability to respond to the resultant variable demand, is responsible for the most significant departure delays. The primary focus of this study is, therefore, to describe and predict how this system behaves over time.

Since it is possible to estimate the probability of being in any particular state at a given point in time for a Markov process, it may be worthwhile to develop a Markovian model for the aircraft departure system. Once the state probabilities have been estimated, it is then possible to determine queue performance measures of interest, such as the expected number of aircraft in the queue and the expected waiting time for a new entry to the queue. Due to the solvability of Markovian models, all of the models developed in this study will have their state spaces defined so that the model is Markovian.

4.2 The Markovian Property

A Markov chain is defined to be a stochastic process for which the conditional probability of transitioning to some future state, given the present state and all previous states, is dependent only on the present state and is independent of the previous states

(16:137). For a continuous time process, this Markovian property applies not only to the conditional probability of transitioning to some future state, but also to the time until the next state transition. The time until the next state transition depends only on the present state of the system and does not depend on how much time the system has been in this state (16:256). These two features of a Markovian model are sometimes described as the memoryless property. A particularly important Markov process is the Poisson process. A Poisson process is a counting process for which the time between events is exponentially distributed (16:211).

Like the Poisson process, the time between state transitions for any Markov process must be exponentially distributed. This is required because the exponential distribution is the only continuous distribution which is memoryless. If the amount of time between events is exponentially distributed, then the amount of time for the next event to occur does not depend on how long it has been since the last event has occurred. In this sense, a memoryless process has no "memory" of anything that has taken place before, such as how long it has been since the last event (16:201).

4.3 Modeling Considerations

For each of the models, the state spaces are defined so that the transition times between states can be modeled with an exponential distribution. This will allow the modeling of the process as a Markov chain which is important to maintaining the tractability of this problem. In order to comply with this requirement, the models developed will apply to periods of time for which the exponential distribution appears to be reasonable. For most of the analysis performed in this study, this time period will be

one hour. The models will be used to solve for the transitory solutions to the state probabilities at the end of each of the time periods to develop a transitory estimate for the non-homogeneous process. The M(t) designation for the entry process in the queue model titles is used to show that the system entry process is Markovian and has a mean arrival rate that varies with time (i.e. non-homogeneous).

LaGuardia has two intersecting runways. The normal operating configuration uses one runway as the primary departure runway and the other as the primary arrival runway. As a result, the models for the aircraft departure process used in this study are all based on a single server system and a single queue. The fact that the runways intersect means that the aircraft arrival process can have an impact on the aircraft departure process. In addition, other factors (GAA departures, traffic from other nearby airports, etc.) will also influence the departure process. Each of the models developed will attempt to account for these factors in one way or another.

The data analysis in Chapter 3 supports the modeling of the aircraft departure process with a Markovian model. Although the number of aircraft that entered the departure queue varied with the hour of the day, when the data was analyzed in short blocks of time, the time between aircraft entries to the queue demonstrated resemblance to the exponential distribution. The analysis further showed that the service process might be reasonably modeled by the exponential distribution. However, it was determined that the service process may be better represented by the Erlang distribution. This turns out to be only a minor hindrance to modeling the system as a Markovian process. Since an Erlang random variable can be represented as a sum of an integer number of independent and

identically distributed exponential random variables, it is sometimes possible to create a Markovian model when the service time distribution follows an Erlang distribution. This is accomplished using the "method of stages" to specify the state space of the model (9:119). This method will be demonstrated in two of the models developed for this study.

The models developed use several variations to characterize the distribution of the service time. The first assumes an exponentially distributed service time. This results in a queue with a non-homogeneous Markovian entry process, a Markovian service process, and a single server (M(t)/M/1). The second model assumes an Erlang-k distribution for the service time, which results in an M(t)/Ek/1 queue, where k is the number of stages of service. The final model adds one additional refinement to the M(t)/Ek/1 queue model by explicitly modeling the times when the server (the runway) is unavailable to service departing aircraft due to outside factors. The period of time when a server is non-available will be called an absence in order to avoid confusion with previous works which calls a similar phenomenon a server vacation.

4.4 Modeling Assumptions

The two key assumptions/conclusions upon which the models of this study are based are:

1. Since LGA typically uses one runway at a time for departures, the departure process can be modeled as a single server queue.

2. The entry process to the departure queue is a non-homogeneous Poisson process.

These assumptions are critical because, without them, it would be extremely difficult to model the system of interest as a Markov process. Given the assumptions above, the final factor to be decided is how to most accurately represent the distribution for the service time.

4.5 M(t)/M/1 Model

This model assumes that the service time distribution is exponentially distributed. Given this assumption, the aircraft departure process can be represented as a Markovian birth and death process where the state of the system is equal to the number of aircraft in the system. The birth rate is determined by the rate at which the aircraft show up at the end of the runway. This process is assumed to be a non-homogeneous Poisson process. The death rate is determined by the average maximum rate at which take-offs occur for the aircraft represented in the system. Since the data set under analysis only includes the eight major airlines, this death rate, or service rate, is only a fraction of LGA's actual departure capacity. This rate should be a function of such factors as the number of GAA take-offs, the number of aircraft landings at the airport, the weather and the congestion of the TCA.

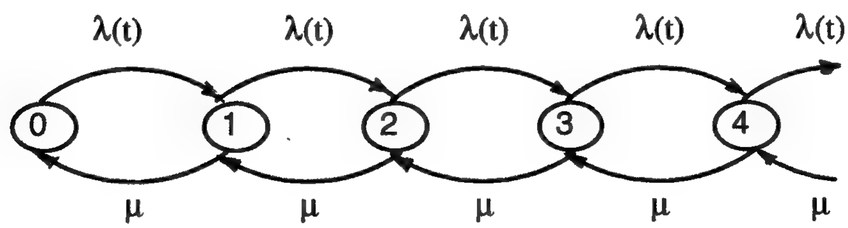


Figure 4.1 M(t)/M/1 Queue Model State Diagram

The state diagram for this model shows that the system description is fairly simple and intuitive. The state of the system corresponds directly to the number of aircraft in the system. The system entry rate is represented by $\lambda(t)$, which stands for the rate of the Poisson arrival process for time period t. The system service rate is represented by μ, which stands for the maximum average rate at which aircraft are allowed to take off.

Both of the required model parameters can be reasonably estimated. The entry rate can be easily estimated by the number of pushbacks (gate departures) scheduled or observed per time period. The service rate is more difficult to estimate. The rate used in this study is the average maximum rate observed for a peak operating period.

4.6 M(t)/Ek/1 Model

This model assumes that the service time distribution follows an Erlang distribution. As was seen in Chapter 3, the Erlang distribution appears superior to the exponential distribution for fitting the service time data. When the first two moments of the empirical data for service times during the AM peak period on 6 June is matched with the theoretical distributions, the Erlang-2 distribution achieves the closest match.

The use of the Erlang-k distribution for the service time greatly increases the flexibility of the model. By varying the value of the parameter k, it is possible to vary the shape of the distribution. In this way, it may be possible to achieve a more accurate representation of the service time. The method of stages uses this Erlang-k representation for the service time distribution to model a system for which the service rate is not exponentially distributed and yet still maintains the advantages of a Markovian model.

However, the state space definition for this type of model is less intuitive than the M(t)/M/1 model. The state of the system does not represent the number of aircraft in the system but rather the number of stages of service in the system.

When using the method of stages to describe the service time, each new aircraft which enters the system can be thought of as bringing k stages of service with him. Thus an entry increases the state of the system by k states, since it will require the completion of k additional stages of service in order to complete service on all aircraft in the system at that time. The state of the system is thus equal to the total number of stages of service that need to be performed in order to complete service on all the customers who are in the system. Each aircraft in the queue will account for k states, while the aircraft "in service" will account for the number of stages of service it has yet to complete. When an aircraft completes one of its stages of service, the state of the system decreases by one. However, the number of aircraft in the system does not decrease until the aircraft in service has completed all k of its stages of service.

The diagram below depicts the state space for such a system where a service has two stages (i.e. k = 2). The decreasing state transition rates in the diagram are shown as 2μ. However, in general each rate will be equal to $k\mu$.

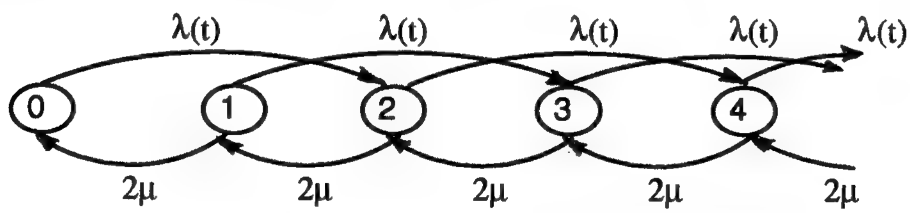


Figure 4.2 M(t)/E2/1 Queue Model State Diagram

Using the method of stages to represent the service time improves the accuracy of the model by allowing a better probability distribution fit for the service times observed. Also, by defining the state of the system as the number of stages of service in the system, the transitions between states remain exponentially distributed and thus the model remains Markovian. Although the model is no longer a birth and death process model, the fact that it is still Markovian greatly simplifies the estimation of the state probabilities.

The estimation of model parameters is almost as straight forward as it was for the M(t)/M/1 model. The entry rates are determined in exactly the same fashion. The service rate μ is also determined in the same fashion as the previous model. However, the transition rate for the completion of a stage of service is equal to $k\mu$. Since k stages must be completed to finish service on one aircraft, the rate at which k stages of service are completed (when each stage has rate $k\mu$) is μ.

4.7 M(t)/Ek/1 Model with Random Server Absences

This model adds one additional refinement to the model just developed. In this model, the server (runway) is periodically unavailable to service the departure aircraft. The server absence can be caused by GAA departures, conflicts with arriving aircraft, or conflicts with other aircraft in the TCA. The previous two models accounted for these factors by using the exponential and the Erlang-k distributions to model the service time. These two distributions have a significant amount of probability mass in their tail, which provides a reasonable representation for the longer than normal "service" times that are sometimes observed. The server absence model can be more intuitively appealing due to

its explicit modeling of the variations in the service time. In this model, the basic service time is still modeled as Erlang-k. However, since the longer than normal service times are represented by server absences, the basic service time has a smaller mean and variance than did the service time distribution from the M(t)/Ek/1 model. This requires the use of an Erlang distribution with a larger parameter k and a smaller mean. The probability of server absences and the rate that the server returns from these absences is used to account for the largest variations in the service times. This model enables the probability of a server absence to be independent of the service rate. The distribution for the amount of time the server is absent is modeled with the Erlang distribution. States are included in the model to represent the stages of the server's return from his absence.

The diagram in Figure 4.3 depicts the state space with two stages of service and two stages of absence return..

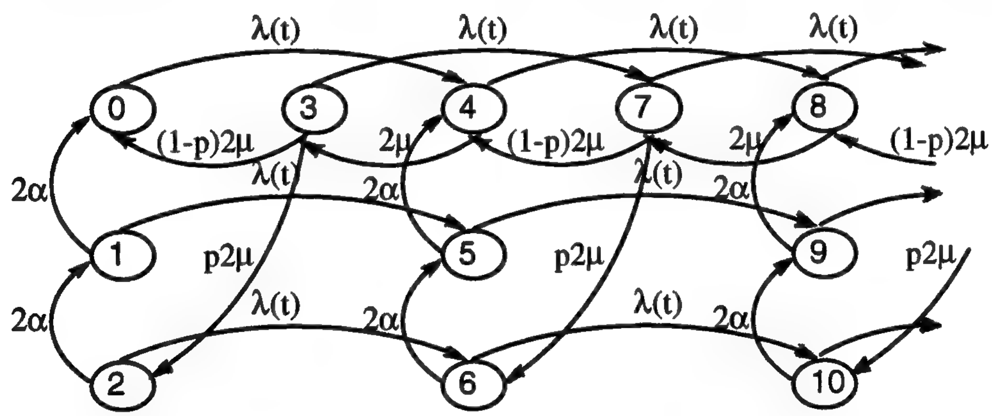


Figure 4.3 M(t)/E2/1 Queue Model With Server Absences State Diagram

In this diagram, the decreasing state transition rates representing absence completions are shown as 2α. The last stage of service completion for each aircraft is a

split Poisson process. The rate when a server absence occurs is equal to $p2\mu$, while the rate where no server absence occurs is $(1-p)2\mu$. The parameter p is the probability that a server absence occurs after a service completion

There are several disadvantages with this latest model. First of all, the increase in the state space can drastically increase the complexity of the problem and thus the amount of computer time required to solve the problem. The second is that the increase in the complexity of the problem requires the accurate estimation of two parameters that may be hard to determine, the probability of a server absence and the absence completion rate. The primary advantage of this model is its explicit modeling of a primary cause of extended departure delays. Even with its limitations, this model generates some interesting results.

4.8 Summary

In this chapter, three models were developed to represent the aircraft departure queue. The first model represents the aircraft departure queuing process as a birth and death process. This model assumes that both the queue entry and the service processes are Markovian. This was shown in Chapter 3 to be a reasonable assumption. The second model retained the requirement that the queue entry process be Markovian. However, the service process was generalized in an attempt to more accurately model the service time. In this model the service time was modeled with the Erlang-k distribution, and the state space was expanded to include k stages of exponentially distributed service. The third model expands on the previous two models by explicitly modeling the periodic increases in the service time with a separate process called a server absence. The probability of a

server absence is assumed to be independent of the service process. Also, the time of each absence is modeled with the Erlang distribution. The state spaces for each model have been carefully designed to preserve the Markovian property of the model. This property will be exploited in the next chapter in order to estimate the system's transitory state probabilities and queue performance measures.

5. Solution Methodology

5.1 Overview

Now that accurate Markovian models have been created, it is possible to estimate the transitory solution for the aircraft departure queuing system. In order to compute this solution, an appropriate time period must be established for which the model parameters appear homogeneous. Most of the analysis in this study uses a time period of one hour. Once this time period is established, model parameters are estimated and the rate matrix is developed for each time period. The rate matrix is used to obtain an approximate solution to the transition matrix. The transition matrix is then used to generate the state probabilities for the end of the time period. The state probability vectors are computed recursively, with the state probability vector for the end of each time period used as the initial condition for the subsequent time period. Finally, these state probabilities are used to generate the desired system performance measures such as the mean and the standard deviation of the number of aircraft in the queue and the expected waiting time in the queue.

The models developed in Chapter 4 are all based on an infinite queue length. However, in order to obtain an approximate solution, each of the models is truncated to a finite dimension. This is justified because the probability that the system is in high states is small enough that truncation does not significantly affect the results. One of the key considerations is determination of an appropriate truncation point.

5.2 Transition Matrix

In order to determine the state probabilities for the end of each time period, it is necessary to determine the transition probabilities for each possible state transition. This is accomplished by solving a series of first order differential equations that represent the rate at which transition probabilities change with respect to time. The number of these equations which need to be solved is equal to the number of state transitions in the system. The Kolmogorov backward equations exemplify one way to set up these equations (16:267).

The transition matrix is the matrix solution of this system of differential equations. This solution represents the transition probabilities for the end of the selected time period. If the subscript i represents the state of the system at the start of the time period and j represents the state at the end of the time period, then the transition matrix element P_{ij} is the conditional probability of being in state j at the end of the time period, given that the system started in i. Once these probabilities have been determined, the resulting matrix is multiplied with the vector of estimated state probabilities for the beginning of the time period to determine the vector of estimated state probabilities for the end of the time period.

The first step in obtaining the solution of these equations is to truncate the number of states in the model to a reasonable level. For this study, a maximum system size is considered reasonable if it is 2 to 2.5 times the magnitude of the highest average queue length generated in the output. When this system size was employed, the state probability for the highest state was always observed to be less than 0.01, and typically it was much

smaller. The second step is to determine the proper rate matrix. This rate matrix is then used in the approximation methods presented by Sheldon Ross to achieve an estimate for the transition matrix (16:291). Both of these approximation methods will be described below. Justification for both approximation methods is provided in Appendix 3.

5.2.1 Rate Matrix

The rate matrix is made up of the transition rates for the model to be solved. If the matrix is called R, then the elements R_{ij} ($i \neq j$) are equal to the transition rates from state i to state j, and the diagonal elements, R_{ii}, are equal to the negative of the total rate at which transitions occur out of state i.

5.2.2 Approximation Method One

The first approximation uses the following identity.

$$P(t) = \lim_{n \to \infty} \left(I + R \cdot \frac{t}{n} \right)^n \tag{5-1}$$

In order to avoid problems with computer errors, n must be chosen large enough so that all of the matrix elements are non-negative prior to performing matrix multiplication. Also, because the approximation only requires the single result of a high order matrix multiplication, this approximation avoids the need to accomplish the many intermediate matrix multiplications. Where other computation methods require n matrix multiplication for each value of n used in the summation, the approximation formula (5-1) allows the repeated multiplication of the matrix product with itself. Thus when $n = 2^k$, the n^{th} power of the matrix may be obtained with only k matrix multiplications.

5.2.3 Approximation Method Two

The second approximation method uses equation (5-2). This equation is a good approximation for the matrix solution, P(t), when n is large.

$$\left[\left(I - R\cdot\frac{t}{n}\right)^{-1}\right]^{n} \tag{5-2}$$

This approximation avoids problems with computer round off error because the matrix $(I - Rt/n)^{-1}$ has only non-negative elements. In addition, this approximation may utilize the same matrix multiplication technique as the first approximation, since only the final matrix product is needed.

The implementation of the two approximations and the computation of the queue performance measures will be demonstrated with their application to the three models of this study.

5.3 M(t)/M/1 Queue Model Solution

For a birth and death process, the transition rates between states are equal to the rates at which entities enter and depart the system. The rates in this model are determined by the rates at which aircraft demand service and the rate at which they are allowed to take off. The rate matrix for this process turns out to be tri-diagonal.

5.3.1 M(t)/M/1 Model Rate Matrix

As an example, a rate matrix is shown in Figure 5.1 for a system which has aircraft entering the system (and demanding service) at a rate of 15 per hour and a service rate of 20 take-offs per hour. The elements (i,i+1) represent the rate at which the state of the system (the number of aircraft in the system) increases, which is equal to the demand for

service rate, 15. The elements (i,i-1) represent the rate at which the state of the system decreases, which is equal to the service rate, 20. Finally, the negative elements on the diagonal (i,i) represent the total rate at which the system transitions out each individual state. Hence, the elements in each row must sum to zero. The matrix, R, below represents a system which is limited to 9 aircraft.

$$R = \begin{bmatrix} -15 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 20 & -35 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 20 & -35 & 15 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 20 & -35 & 15 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20 & -35 & 15 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 20 & -35 & 15 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 20 & -35 & 15 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 20 & -35 & 15 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & -35 & 15 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & -20 \end{bmatrix}$$

Figure 5.1 Rate Matrix -- M(t)/M/1 Queue Model

The boundary conditions require special consideration. The process obviously cannot transition to a lower state when the state is 0, so element (0,0) is -15. Also, since the truncation point in this example is 9 aircraft, aircraft arrivals are not allowed when there are 9 aircraft already in the system. Thus, element (9,9) is -20.

5.3.2 M(t)/M/1 Model State Probabilities

Now that the rate matrix has been determined, the length of the time period, t, and the value of n must be established. For the above example, if t equals 1 hour, n must be greater than 35 in order to avoid negative elements in the matrix (I+Rt/n) used in the first approximation method. In this case, the lowest acceptable value of k is 6, which yields an

n of 64. Although the second approximation method does not have the same problems with negative elements, n must still be large in order to obtain an accurate approximation.

Once the transition matrix has been approximated, it is multiplied with the state probability vector for the system condition at the beginning of the time period to obtain the approximation for the state probability vector at the end of the time period. At LaGuardia, the airport is closed from midnight until 6:00 AM. Therefore the initial condition state probability vector has a state 0 probability equal to 1.0 and all other state probabilities equal to 0.0. In general, however, any initial system state may be used. For example, if the model were to be run with an initial state of 2 aircraft in the system, then its initial state vector would have the probability 1.0 in state 2 and probability 0.0 elsewhere.

5.3.3 M(t)/M/1 Model Queue Performance Results.

The queue performance measures are determined once the state probabilities have been estimated. The expected queue length is found by summing of the products of each state probability and the queue length associated with that state. When the system is in state i, there are i aircraft in the system, and (i-1) aircraft in the queue. Equation (5-3) below demonstrates how the expected queue length may be determined.

$$Nq = \sum_{i=1}^{max} (i-1) \cdot Pn_i \qquad (5\text{-}3)$$

Nq represents the expected number of aircraft in the queue. Pn_i represents the estimated probability of being in state i. The variable "max" represents the maximum system size as determined by the chosen truncation point.

The variance of the number of aircraft in the queue is estimated by calculating the expected value of $(i-1)^2$ and then subtracting $(Nq)^2$ from it.

$$Vq = \sum_{i=1}^{max} (i-1)^2 \cdot Pn_i - (Nq)^2 \tag{5-4}$$

The next queue performance measure computed is the expected waiting time for an aircraft which enters the system at the end of a time period. This waiting time is estimated by computing the average amount of time it will take to complete service on all aircraft in the system at the time the new aircraft shows up. When i aircraft are in the system, a new aircraft can expect to wait i times the average service time. Since the state of the system is equal to the number of aircraft in the system, this amount is multiplied by the probability of being in state i. The results are computed and summed for all possible states. This calculation is accurate even when one of the aircraft in the system has been in service for a while. Since the service time is Markovian, it is memoryless and so the expected time remaining for an aircraft already in service is the same regardless of how long it has already been in service.

Equation (5-5) demonstrates how this virtual waiting time is computed when the average service time is represented by $1/\mu$.

$$Tq = \sum_{i=1}^{max} Pn_i \cdot \frac{i}{\mu} \tag{5-5}$$

5.4 M(t)/Ek/1 Queue Model Solution

This model allows more flexibility in the representation of the service time distribution. But, since servicing is accomplished in stages, the process is no longer a birth and death process and the rate matrix is no longer tri-diagonal. In order to determine the appropriate rate matrix, the parameters for the service time distribution must be accurately estimated. The method used in this study was to match the first two moments of the Erlangan-k distribution to the data for a peak period when significant delays were encountered. This procedure was performed on the data from 6:00 to 9:00 AM on 6 June and yielded an Erlang-2 distribution.

5.4.1 M(t)/Ek/1 Model Rate Matrix

As an example, a rate matrix is shown in Figure 5.2 for a system which has aircraft entering the system at a rate of 15 per hour and a service rate of 20 take-offs per hour and 2 stages of service.

$$R = \begin{bmatrix} -15 & 0 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 40 & -55 & 0 & 15 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 40 & -55 & 0 & 15 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 40 & -55 & 0 & 15 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 40 & -55 & 0 & 15 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 40 & -55 & 0 & 15 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 40 & -55 & 0 & 15 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 40 & -55 & 0 & 15 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 40 & -55 & 0 & 15 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 40 & -40 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 40 & -40 \end{bmatrix}$$

Figure 5.2 Rate Matrix -- M(t)/E2/1 Queue Model

In this example, the maximum number of aircraft in the system is set to 5. This results in a matrix dimension of 11 since each of the 5 aircraft can have two stages of service, which accounts for 10 of the states. The one additional state is the 0 state.

This rate matrix demonstrates how this type of system must increase k states at a time. In this model, an aircraft can be thought of as bringing k stages of service with it when it enters the system. The system state still decreases in increments of one since service is completed one stage at a time. The boundary conditions and the row total requirements are similar to those discussed for the M(t)/M/1 queue.

5.4.2 M(t)/Ek/1 Model Queue Performance Results

Now that the rate matrix has been determined, it is possible to calculate the transition matrix and then the estimated state probabilities using the same methods as the M(t)/M/1 model. However, the queue performance measure calculations are slightly more difficult. In the equations which follow, "max" still represents the maximum number of aircraft in the system and i represents the number of stages of service that the aircraft presently in service has remaining. The subscript j represents the number of aircraft in the system. Equation (5-6) shows how the expected number in the queue may be determined.

$$Nq = \sum_{j=1}^{max} (j-1) \cdot \left[\sum_{i=1}^{k} Pn_{k(j-1)+i} \right] \tag{5-6}$$

In this case, each k adjacent states above the 0 state represent the same number of aircraft in the system. When the system is in state k*(j-1)+i, there are j aircraft in the system and (j-1) aircraft in the queue.

The calculation of the variance of the queue length is similar to that used for the M(t)/M/1 model and is demonstrated by equation (5-7) below.

$$Vq = \sum_{j=1}^{max} (j-1)^2 \cdot \left[\sum_{i=1}^{k} Pn_{k(j-1)+i} \right] - (Nq)^2 \tag{5-7}$$

Finally, the calculation of the expected waiting time is demonstrated with equation (5-8). Average service time per aircraft is represented by $1/\mu$.

$$Tq = \sum_{j=1}^{max} \sum_{i=1}^{k} Pn_{k(j-1)+i} \cdot \frac{k \cdot (j-1) + i}{k \cdot \mu} \tag{5-8}$$

When the average service time is $1/\mu$, and there are k stages of service, average time to complete each stage of service becomes $1/k\mu$.

5.5 M(t)/Ek/1 Server Absence Model Solution

This model continues to use the Erlang representation of the service times. However, it also includes the explicit modeling of occasional server absences. In this model, the service time is distributed more Gaussian and thus has a much smaller variance than the models without explicit modeling of server absences. Therefore, it is important to use the Erlangan distribution to capture this much greater central tendency. The excursions from this service time distribution are modeled as a separate (server absence) process. The amount of time the server is absent is modeled with an Erlang distribution. The increase in the level of complexity for this model greatly increases the size of the state space and the complexity of the rate matrix. As a result, the solution of this model is much more computationally intensive.

5.5.1 M(t)/Ek/1 Server Absence Model Rate Matrix

In the M(t)/M/1 model, each state represented a different number of aircraft in the system. The M(t)/Ek/1 model imposes almost a k-fold increase in the state space. This final model, the M(t)/Ek/1 server absence model, increases the size of the state space by almost $(k_1 + k_2)$ fold over the M(t)/M/1 model. The parameter k_1 represents the number of stages of service, while k_2 represents the number of stages involved in the server's return from his absence. The following rate matrix is an example of the numerous types of state transitions which can occur. For this example, there are 2 stages of service and 2 stages of return for a server absence. The rate that aircraft enter the system is 5, the service rate is 10, the absence probability is 0.2 and the absence return rate is 20. The maximum number of aircraft in the system is 2. The resulting matrix dimension for this example is (k_1+k_2)*max + k_2 +1, or 11.

$$R = \begin{bmatrix} -5 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 40 & -45 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 40 & -45 & 0 & 0 & 0 & 5 & 0 & 0 & 0 & 0 \\ 16 & 0 & 4 & -25 & 0 & 0 & 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 20 & -25 & 0 & 0 & 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 & 40 & -45 & 0 & 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 0 & 0 & 40 & -45 & 0 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 16 & 0 & 4 & -20 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 20 & -20 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 40 & -40 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 40 & -40 \end{bmatrix}$$

Figure 5.3 Rate Matrix -- M(t)/E2/1 Queue Model with Random Server Absences

The diagonal of 5s represent increases in the state space due to aircraft entering the system. This shows that each entering aircraft increases the state of the system by

(k_1+k_2) states (4 states in the example). The intermittent diagonal of 16s represent the rate at which the final stage of service is completed on each aircraft when an absence does not occur immediately following it. This is determined as the product of the probability that an absence does not occur and the rate of a stage of service, or $(1\text{-}0.2)*k_1*10 = 16$. The elements with the value 4 represent the rate at which the final stage of service is completed on each aircraft when an absence does occur immediately following it. This is equal to $0.2*k_1*10 = 4$. The elements with the value 40 represent the rate at which the stages of the absence are completed, which is equal to k_2*20. The elements with the value 20 represent the rate at which stages of service are completed for stages other than the last one, which is equal to k_1*10. Finally, the diagonal elements, R_{ii} , are the negative of the total of the all the transition rates out of state i.

As is evident from this example, the size of the rate matrix and its complexity has greatly increased. Nevertheless, the estimation of the state probabilities may still be accomplished with the approximation methods used in this study. One of the major disadvantages of this model turns out to be the large amount of computer time required to solve it. This issue will be addressed in Chapter 6.

5.5.2 M(t)/Ek/1 Server Absence Model Queue Performance Results

As the complexity of the rate matrix would indicate, the computation of the queue performance measures is more difficult than the previous two models. In this model the first (k_2+1) states represent 0 aircraft in the system. After these "0 states," each (k_1+k_2) states represent another aircraft. The formula for the expected queue length is Equation (5-9).

$$Nq = \sum_{j=1}^{max} (j-1) \cdot \left[\sum_{i=1}^{(k_1+k_2)} Pn_{(k_1+k_2) \cdot j - k_1 + i} \right] \tag{5-9}$$

Whenever there are j aircraft in the system, there are (j-1) aircraft in the queue. The probability of having j aircraft in the system is equal to the sum of the (k_1+k_2) adjacent state probabilities which represents that number of aircraft.

The estimation of the variance is accomplished with a formula similar to (5-7).

$$Vq = \sum_{j=1}^{max} (j-1)^2 \cdot \left[\sum_{i=1}^{(k_1+k_2)} Pn_{(k_1+k_2) \cdot j - k_1 + i} \right] - (Nq)^2 \tag{5-10}$$

Finally, the estimation of the waiting time is a bit more involved since future server absences will increase the expected time to complete service. In equation (5-11), μ is the service rate, α is the absence return rate, and p is the probability of a server absence at the completion of a stage of service.

$$Tq = \left[\sum_{j=1}^{max} \left[\sum_{i=1}^{k_1} Pn_{(k_1+k_2) \cdot j - k_1 + i} \cdot \left[\frac{k_1 \cdot (j-1) + i}{k_1 \cdot \mu} + \frac{(j-1) \cdot p}{\alpha} \right] + \sum_{i=1}^{k_2} Pn_{(k_1+k_2) \cdot j + i} \cdot \left[\frac{j}{\mu} + \frac{i}{k_2 \cdot \alpha} + \frac{(j-1) \cdot p}{\alpha} \right] \right] \right] \tag{5-11}$$

(a) (b) (b1) (b2) (c) (c1)(c2) (c3)

The summation over variable j (a) represents the number total number of aircraft that can be in the system. The first summation over variable i (b) represents those states for which the server is present. The first term in this summation (b1) represents the expected amount of time to complete all the stages of service for the state of interest. The second term in this summation (b2) represents the expected amount of delay due to future server absences. The second summation over variable i (c) represents those states for

which the server is absent. The first term in this summation (c1) represents the expected amount of time to complete all the stages of service for the state of interest. The second term (c2) represents the amount of time required to complete the amount of absence time remaining. The third term (c3) represents the expected amount of delay due to future server absences.

5.6 Computer Implementation

The methodology described in this chapter was coded into FORTRAN for implementation. The programs first perform the automated creation of the rate matrix from user input system parameters and rates. The programs then perform the transition matrix estimation algorithm, compute the estimated state probabilities and finally compute the queue performance measures.

Four computer programs were written. The first two compute the solution for the M(t)/Ek/1 model using the two estimation techniques presented. The second two programs handle the M(t)/Ek/1 model with random server absences for the two estimation techniques. The M(t)/M/1 model solution is obtained using one of the M(t)/Ek/1 model programs and setting the parameter k equal to one. The programs which implement the solution using approximation method two call an IMSL library matrix inversion subroutine (8:1130).

All programs output expected queue length, queue length standard deviations, and expected waiting times in minutes. These queue performance measures, the rate matrix and the state probability matrix are written to separate output files. The FORTRAN code and instructions on how to implement them are included in Appendix 4.

5.7 Conclusion

The primary feature of the departure queue at LaGuardia Airport is its transitory nature. Therefore, the system was represented with Markovian models in order to facilitate the estimation of transitory state probabilities and queue performance measures. The two estimation techniques are used to solve for the transition matrix for the departure queue models. These results were then used to estimate state probabilities and queue performance measures. Chapter 6 presents the results for how these models performed for LaGuardia's departure queue for the seven days of the study.

6. Results

6.1 Overall

Each of the models developed generates estimates for the system state probabilities for the end of each time period. These results provide the user with a large degree of flexibility. Although the primary system performance measures for this study are the expected queue length and the expected waiting times, the state probabilities may be used to obtain a host of other measures of interest. For example, if the user were interested in the probability of the airport experiencing a queue length greater than a certain amount, he could easily determine this using the state probabilities generated.

The exponential and the Erlang models were executed with the days for which adequate data existed. These days were 3 through 7 June. The absence model was executed only for 6 and 7 June, because these were the days for the which absences had the most significant effect. The outputs of these models were plotted and compared to the actual roll-out times experienced at LGA for those days.

The 6^{th} and 7^{th} of June were the only days of the study which had any significant amount of time with weather worse than Visual Meteorological Conditions (VMC). They were also the days which experienced the most significant patterns of departure delays. Therefore, the data from these days will be used as the primary basis to evaluate the models. The complete set of plots may be found in Appendix 5.

In order to calibrate each model, the output had to be correlated to the type of data recorded at the airport. One of the models' primary output measures is the time-

dependent queue length. However, queue lengths are not available in the data set. Therefore, the models' output also generates estimates for the expected waiting time of an aircraft that enters the system at the end of a time period. This expected waiting time is then combined with an estimate for the time spent between pushback and the time an aircraft shows up at the queue to determine an estimate for the roll-out time. Since roll-out time is readily obtained from the data set, it is the primary performance measure used to correlate the model output to the airport's actual performance.

In order to determine the best fit of model output to the data, the models were run at various effective service rates and the results compared to the observed data. The results of this calibration are demonstrated below using the exponential service time model, the Erlang service time model and the absence model.

6.2 Exponential Model Results

LGA operated on runways 22/13 for the majority of the day on Monday, 6 June. The 22/13 configuration is listed in the FAA data dictionary as the preferred configuration for LGA (4:2). This is reflected in the Quality Performance Measurement data by the fact that this configuration allows one of the greatest take-off rates (39 per hour). However, LGA experienced some of its most significant delays on this date. The recorded weather and the LGA data dictionary provide a possible explanation.

The surface winds on 6 June varied from 150 to 180 degrees. The data dictionary reveals that LGA may experience departure delays when departing runway 13, if Kennedy International Airport is performing a special type of approach to runway 13L (4:4). Although this could not be confirmed, the wind direction and the weather support the

existence of this condition. This might explain why LGA experienced a low effective service rate on 6 June when its departure capacity was reported to be one of LGA's highest.

In Chapter 3, it was determined that the observed roll-out time during non-peak periods could be modeled as a normal distribution with a mean of 13 minutes. The exponential model output for this same non-peak period, using an effective service rate of 24 take-offs per hour, resulted in an average wait of 5 minutes. Therefore, the difference (13 - 5 = 8) was used as the nominal taxi time. The service rate of 24 take-offs per hour was determined during model calibration. This was the rate which resulted in queue performance patterns which most closely matched those observed on that day.

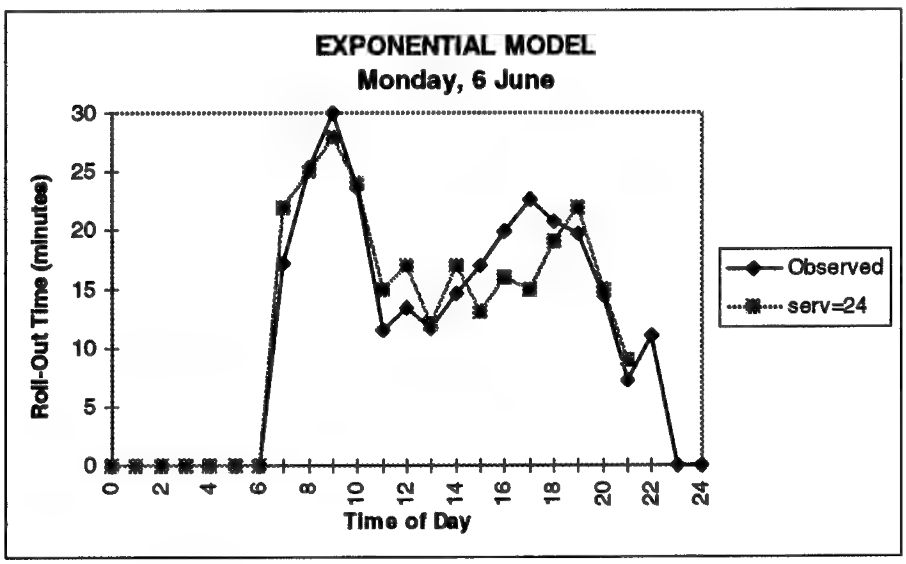


Figure 6.1 Exponential Model Results -- 6 June

This model demonstrates a fairly good fit to the data when an effective service rate of 24 take-offs per hour is used. However, it does appear that the effective service rate may have dropped below 24 between 3:00 and 5:00 PM.

LGA was again operating on runway 22/13 on 7 June from 6:00 to 7:15 AM. However, the wind direction in the morning varied from 180 to 220 degrees. This wind direction may indicate that Kennedy was not operating on runway 13 and thus not performing the type of approaches which interfere signficantly with LGA. If this were the case, it would be expected that the actual effective departure capacity would correlate more closely with the reported capacity. The reported departure capacity for 7 June was greatest for the first hour and a half. The reported capacity was lower for most of the remainder of the day. This may explain why the effective rate appears to start out high and then stabilizes at a lower amount. A service rate of 25 take-offs per hour generates the best fitting exponential model output for the majority of the day.

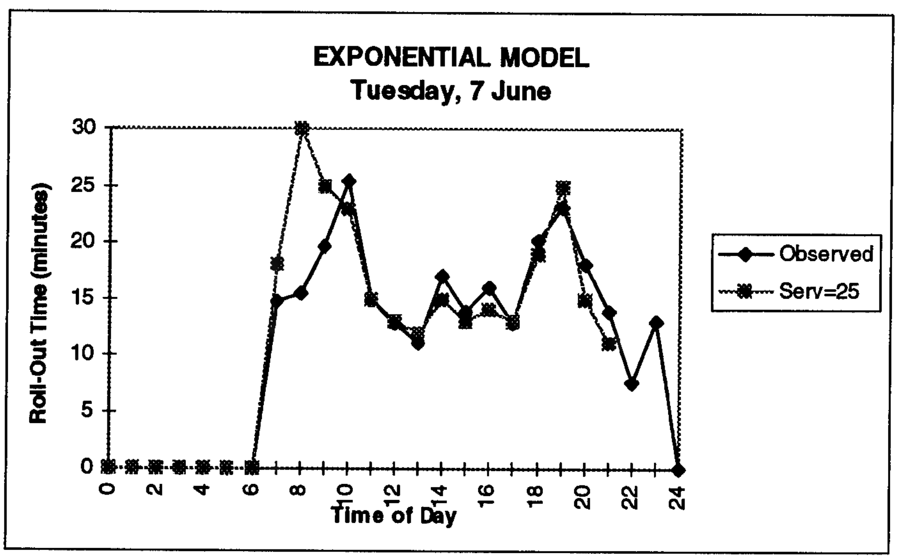


Figure 6.2 Exponential Model Results -- 7 June

This plot demonstrates good correlation to the roll-out times actually observed except for the period from 6:00 to 8:00 AM. As mentioned before, the reported weather and departure capacities were worse on 7 June than they were on 6 June, yet the model calibration indicates that the effective service rate was greater on 7 June. This may be due

to other outside influences (e.g. interference from aircraft flying approaches at Kennedy on 6 June). Overall, the exponential model demonstrates good correlation to the roll-out times actually observed for 6 and 7 June.

6.3 Erlang-2 Model Results

The Erlang model was executed for 6 June using several different service rates. The output using an effective service rate of 23 take-offs per hour generated the results which most closely matched the airport's performance on that day. For the Erlang models, the average time between pushback and queue entry was estimated to be 10 minutes.

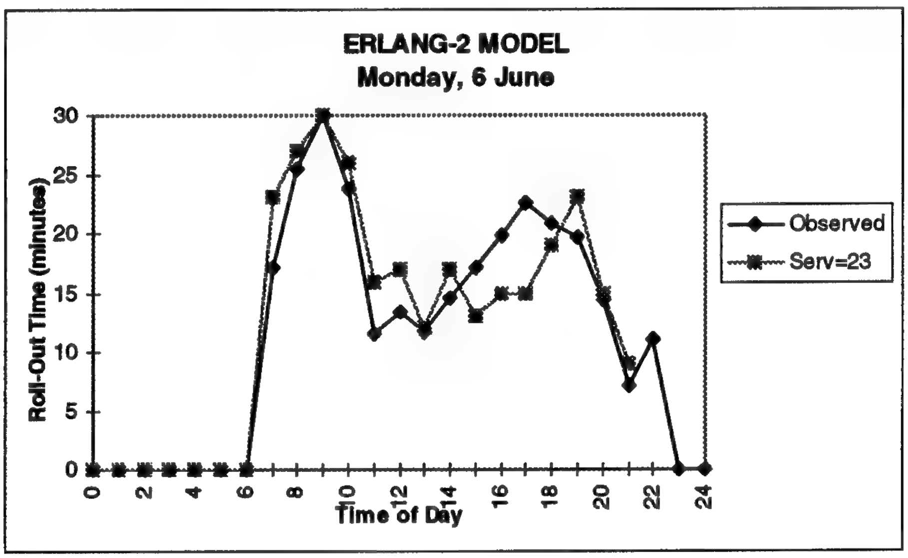


Figure 6.3 Erlang Model Results -- 6 June

This graph demonstrates a similar ability to correlate variations in the roll-out time observed to the time-variant take-off demand process.

The Erlang-2 model results from 7 June use an effective service rate of 26 take-offs per hour.

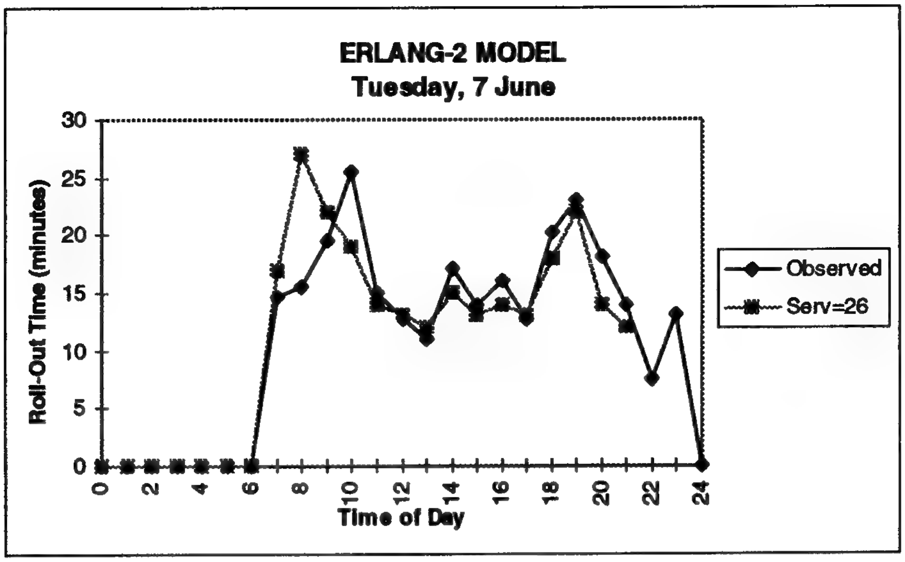


Figure 6.4 Erlang-2 Model Results -- 7 June

The Erlang-2 model demonstrates a good fit for the data on 7 June. Once again the effective service rate appears to have been somewhat greater than 26 take-offs per hour for the first two hours of the day.

6.4 Absence Model Results

This model was performed for 6 June assuming an overall service rate of 35 take-offs per hour, which is close to that reported for this day. This model does not use a fractional service rate because the periods of time when general aviation aircraft are using the runway are modeled separately as a server absence. The probability that the runway would not be available to service an aircraft in this queue was estimated to be 0.2. In Chapter 3, it was shown how this value was estimated from the peak period observations on that day.

This model's output for 6 June demonstrates a reasonable fit to the data. Although a single probability for an absence was used for the entire day, the actual

probability should vary just as the demand rate does. Determining these variable rates is beyond the scope of this study.

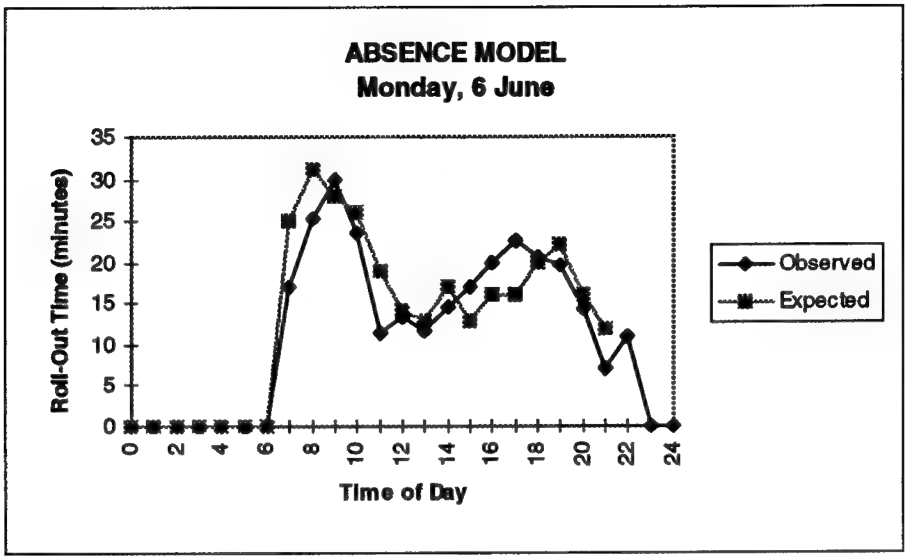


Figure 6.5 Absence Model Results -- 6 June

The absence model for 7 June also used an overall service rate of 35 take-offs per hour. However, the probability of an absence was estimated to be 0.23.

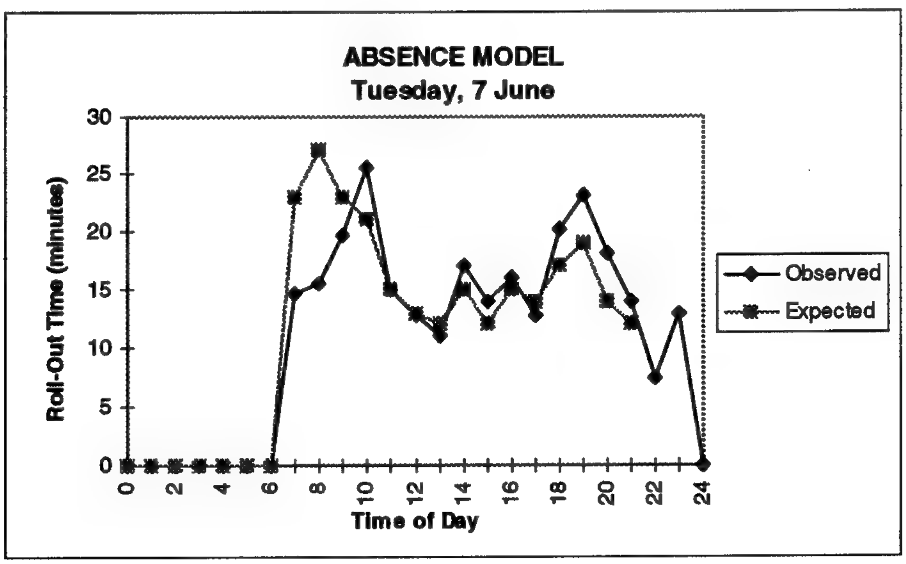


Figure 6.6 Absence Model Results -- 7 June

Although the absence model has an intuitive appeal, it requires the estimation of hourly probabilities of absence in order to generate results that are any better than the exponential and the Erlang models. If the statistical relationship between these probabilities of absence and the effective service rate could be determined, the absence model might turn out to be the more accurate model.

6.5 Service Distribution Comparison

In order to observe the effect of changing the number of stages of service, the Erlang model was executed with 1, 2 and 4 stages of service. The runs were performed with the 6 June data and a service rate of 24 take-offs per hour.

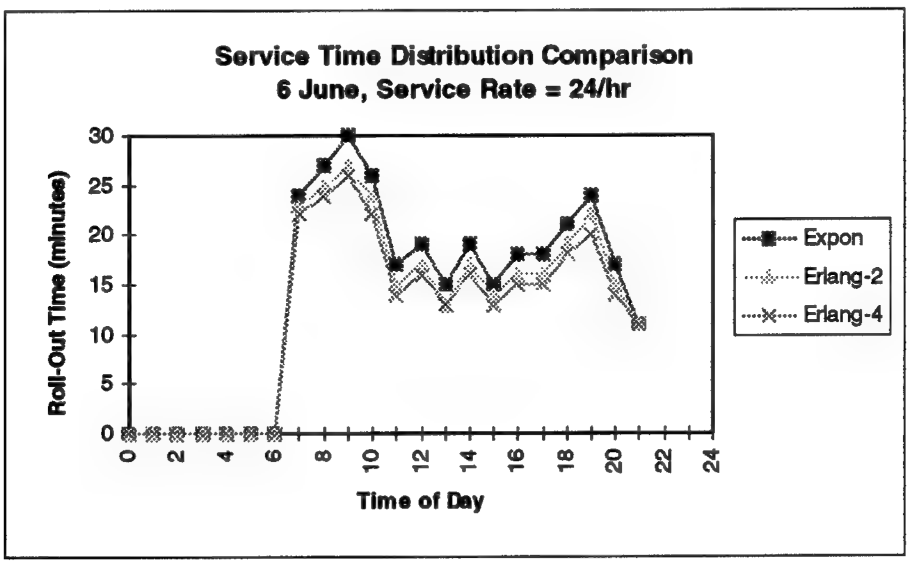


Figure 6.7 Service Time Distribution Comparison

This plot demonstrates that the primary effect is a reduction in the variability of the output results. This should be expected since the variance of the service time distribution decreases as the number of stages of service increases.

6.6 Alternative Time Period Lengths

All three models have the ability to use time periods of various lengths. To demonstrate this capability, the pushback data was determined for 0.5 hour time blocks. These rates were then used as the input for the Erlang model

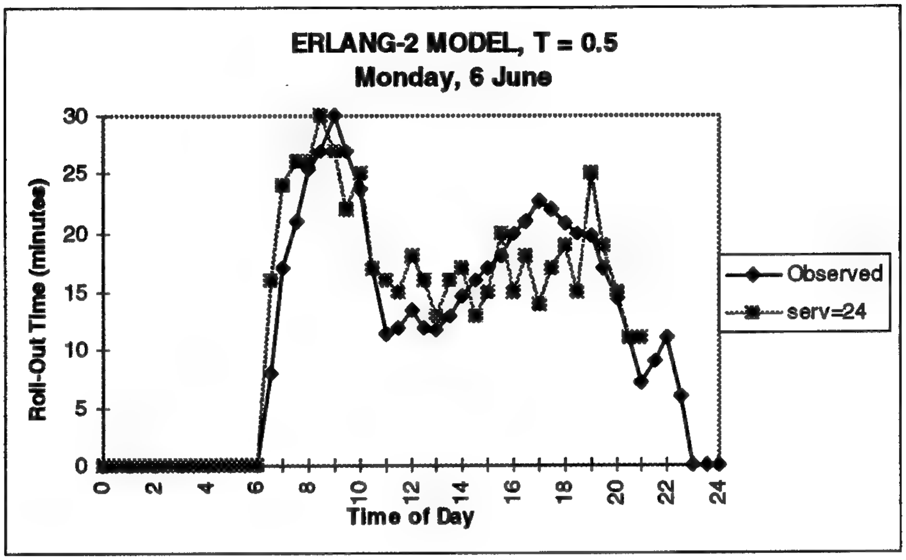


Figure 6.8 Erlang-2 Model Results -- Time Period = 0.5 hour

This result generates a much more erratic graph. However, this should not be interpreted as a lack of precision. In fact, reference to the actual roll-out time plots in Appendix 1 indicates that the roll-outs themselves varied significantly about the local average. This model indicates that much of this variability may be due to the highly variable demand process.

6.7 Computer Code Execution

A listing of the FORTRAN computer code and a user's guide is provided in Appendix 4. Four programs are included: Two for the Erlang model and two for the

absence model. The Erlang model and the absence model each have separate programs to employ each of the two approximation methods.

The computer code developed employs a very simple algorithm for the calculation of the matrix products. This algorithm performs every multiplication and does not take advantage of any special characteristic of the matrices. It may be possible to improve computation times if a more efficient matrix multiplication subroutine can be found.

The programs which use the approximation method with matrix inversions call an IMSL subroutine. Surprisingly, the implementation of the models with the matrix inversion does not noticeably increase the computation time over that experienced for the non-inversion models. The matrix multiplications appear to be responsible for the largest portion of the computation time.

The programs were executed on a UNIX, SPARC-20 computer terminal. As was expected, computation times were found to be quadratically related to the dimension of the rate matrix. Several executions were performed with gradually increasing matrix dimensions. The results are graphed in Figure 6.9.

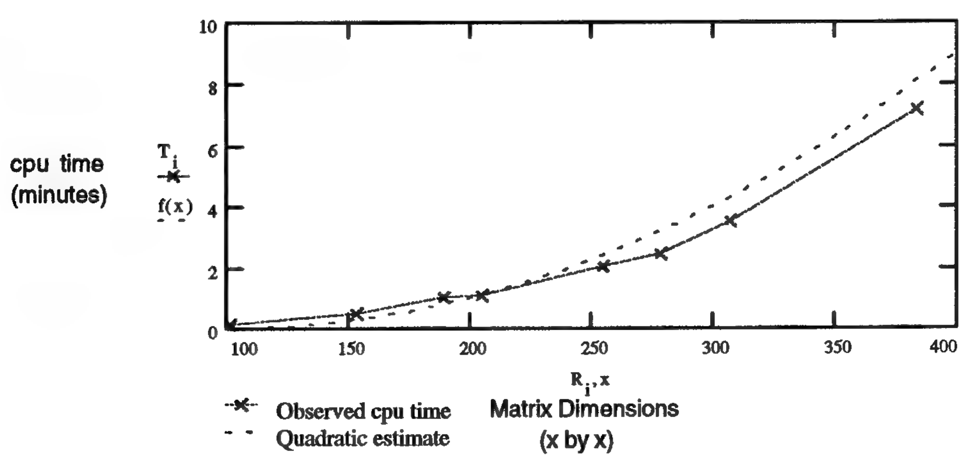


Figure 6.9 Computation Times

These results were generated using one time period of data, and an accuracy factor of 6 (meaning that six matrix multiplications were performed for one run). For the Erlang-2 models with a maximum queue length of 25, the resulting matrix dimensions were 52 X 52. For this model, 15 time periods (hours) of data took about 20 seconds of cpu time. However, as demonstrated in the graph above, computation times become a problem for the increased dimensionality of the typical absence model. The two executions of the absence model for this study had matrix dimensions of 385 X 385. It took approximately 7.5 minutes for one time period and approximately two hours to complete 15 time periods of data.

6.8 Conclusion

All of the models in this study demonstrated good ability to relate the variable demand rates to the delay patterns actually observed at LGA. The exponential model assumes a Markovian service process and thus models this process with larger variance than actually observed. The Erlang model uses a probability distribution which matches the actual service process more closely. The absence model attempts to explicitly model the variance in the service process and has the potential to achieve the closest fit for the service time distribution. However, the absence model proved to be much more difficult to employ and much more time consuming than the Erlang model.

7. Conclusions

7.1 Summary

The purpose of this research was to develop an improved model for the aircraft departure process at an airport. The model developed was to be used to improve the representation of the departure delay process. It was also to be used to improve the ability to predict the occurrence of delays.

Initial data analysis indicated that the departure delay process was non-stationary. The primary factor which influenced the occurrence of these delays was determined to be the time variant demand rate. Although this demand process was non-stationary, it was determined that it could be reasonably modeled as a homogeneous Markovian process when analyzed in short periods of time (one hour or less).

The model was developed with data from LaGuardia airport, which typically uses one runway for arrivals and one for departures. Therefore, the departure process was assumed to have a single server. The service process was found to be reasonably modeled as a Markovian process. However, it was determined that the service process was more accurately represented with an Erlang distribution. As a result, the method of stages was used in modeling the service process. This enabled a closer probability distribution fit for this service process and allowed the model to be Markovian. The use of a Markovian model was important for calculating the transitory solution to the non-stationary process.

Three models were developed and analyzed: the exponential model, the Erlang model and the absence model. The Erlang model is the recommended model. Its results

correlate well with the actual delays observed at LGA. In addition, it has the flexibility to fit the actual service time distribution more closely. Finally, it generates results quickly and efficiently.

One of the primary advantages of the Erlang model developed is its flexibility in representing the service times. If the data set had been more complete and included all scheduled aircraft departures, the service time representation could easily be modified to accurately model this process. In this case, the service process would not experience as much variation because the GAA departures might be included in the data set. Also, the resulting service process should take on a more Gaussian shape. This is easily achieved with the Erlang model by increasing the number of stages of service. The flexible service time representation afforded by the Erlang model might also enable this type of model to be used at classes of airports different from LaGuardia. Current models do not provide this flexibility for service time representation.

Another advantage of using the Erlang model for departure capacity estimation is that it allows the user to see periods of time when the effective service rate varies. It also smoothes out the results over time. This helps to avoid over estimation of capacity when the surges in effective service rate occur. Some methods for capacity estimation require identifying when these surges occur and then ignoring those observations (6:146).

The models developed in this study may also have applications for a single runway configuration where departures and arrivals utilize the same runway. In this case, the periodic preemption of the runway by blocks of landing aircraft could be accurately

represented by increasing the service time variance in the Erlang model, or increasing the probability of a server absence in the absence model.

7.2 Applications

It is believed that the models developed in this study could be used to improve estimation of the effective service rate for the set of aircraft under study. The correlation of these rates to factors such as the number of GAA departures, the weather, the operating configuration, etc., would allow the use of these models to aid in the prediction of departure queues and departure delays at an airport.

Whereas the determination of effective service rates was accomplished using the actual pushback (gate departure) times, delay prediction could be accomplished using the Official Airline Guide (OAG) scheduled pushback times. In the absence of properly correlated effective service rates, the best fitting effective service rate could be determined for an hour just completed. This rate might then be used to estimate the effective service rate for the next hour.

At the present time, departure delay prediction is virtually non-existent. A delay is only predicted if an aircraft has not departed by five minutes after its expected take-off time. The expected take-off time is determined by adding an estimated taxi-out time to the OAG scheduled gate departure (pushback) time. If the aircraft has not taken off at its current predicted take-off time, this time is incremented five minutes. The process continues until the aircraft takes off or an hour has passed. At one hour past the original time, the aircraft is deleted from the system. If the aircraft subsequently takes off, it is

reentered into the system. (21:1) The models developed in this study may be useful for improving the process.

Finally, it is worth mentioning several other model applications. Since the output includes the probability distribution for the state vector at the end of each time period, it is possible to estimate the probability of long queues. Using the Erlang model output from 6 June, and an effective service rate of 23 take-offs per hour, there is a 10 % chance that the queue length will be greater than 17 aircraft at 9:00 AM. This type of analysis may be useful in helping to avoid extremely "ugly" departure queues.

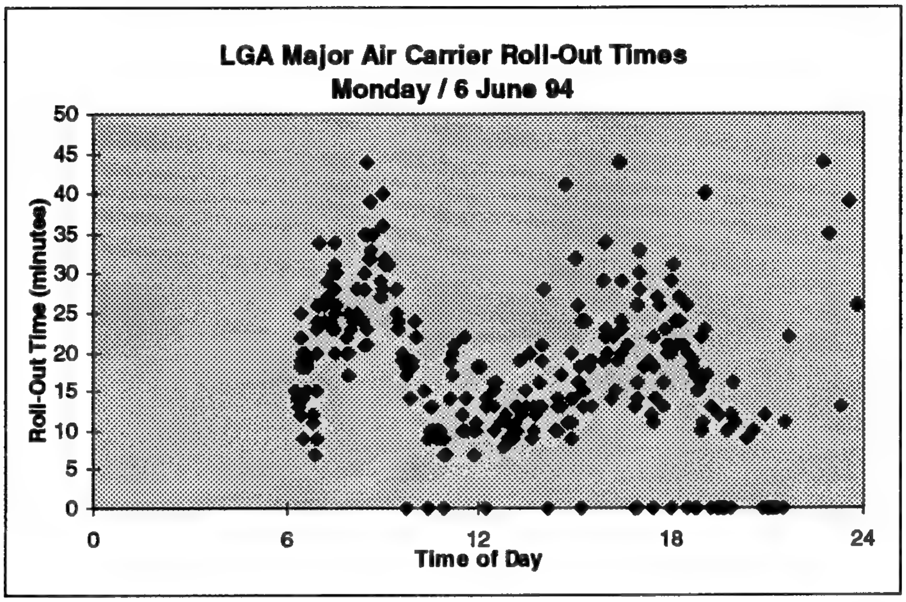
Another application may be to determine what effect opening the airport earlier and scheduling departures more uniformly would be. Again using the Erlang model results from 6 June, if the airport were opened one hour earlier and the number of pushbacks (gate departures) were scheduled more uniformly and limited to 20 per hour, the expected mean waiting time remains below 22 minutes. In addition, there is a 90 % probability that the queue length will remain below 11 aircraft.

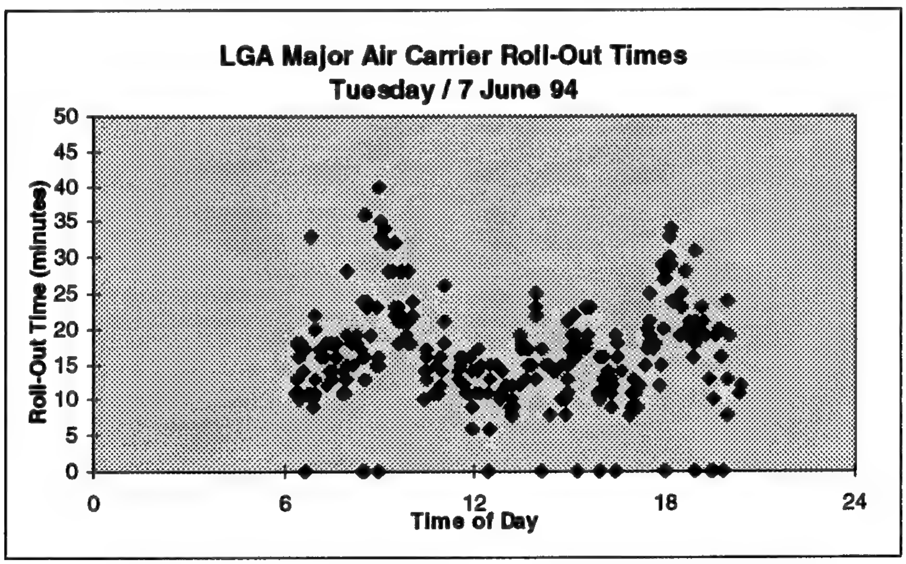
7.3 Recommendations for Further Research

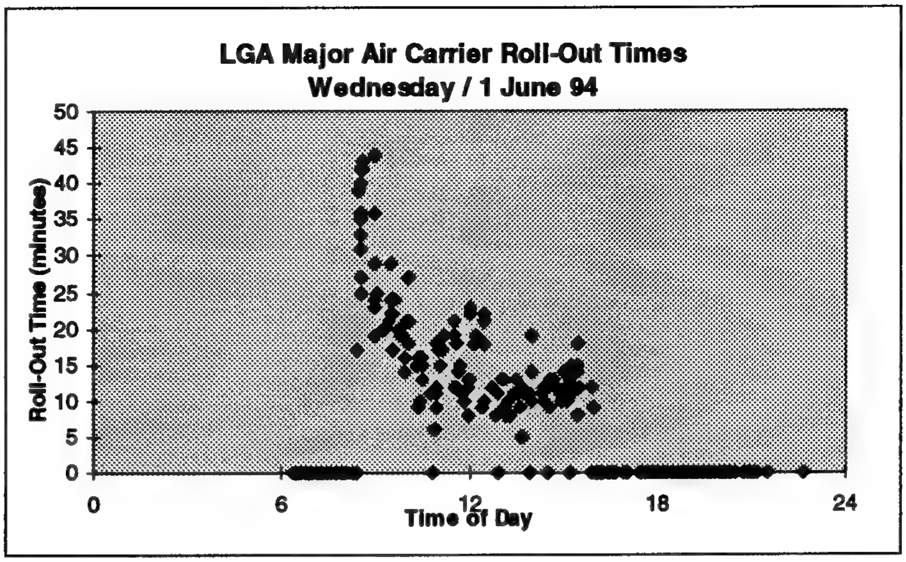
This study was performed with a limited data set. There were only two days which experienced significant delays. With more days of data it may be possible to determine the statistical significance of the various factors which contribute to reduced effective service rates, and consequently, departure delays. In addition, further research could be performed to determine the applicability of these models to larger airports and different runway configurations.

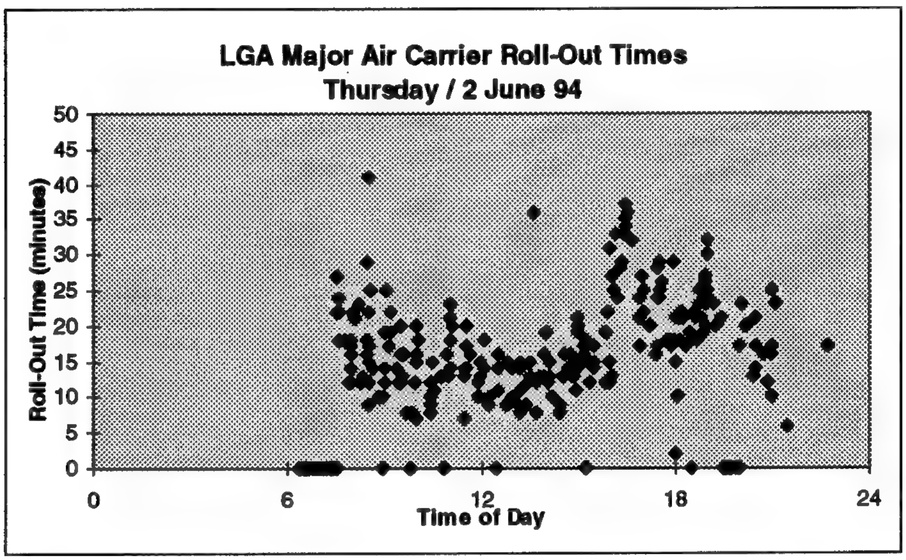
Appendix 1: Data Plots

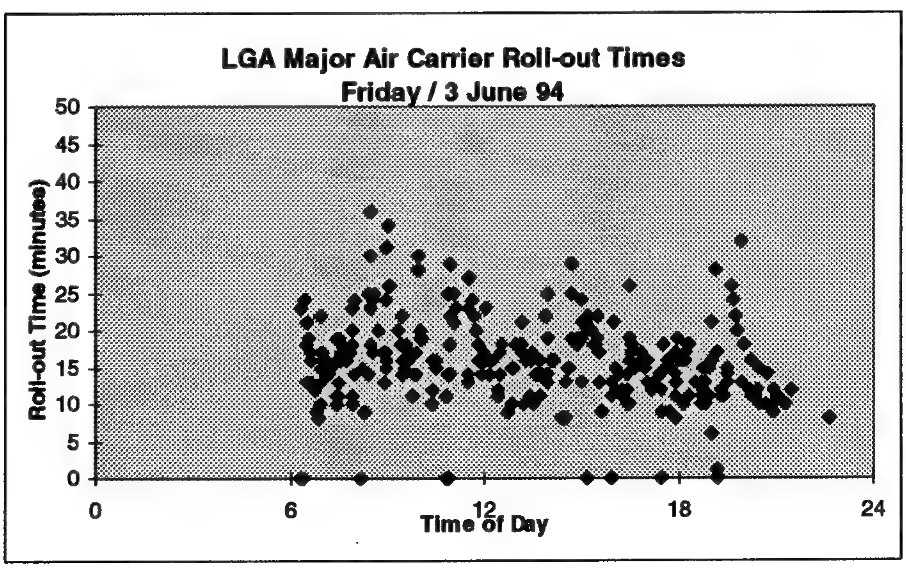
Roll-Out Time Plots

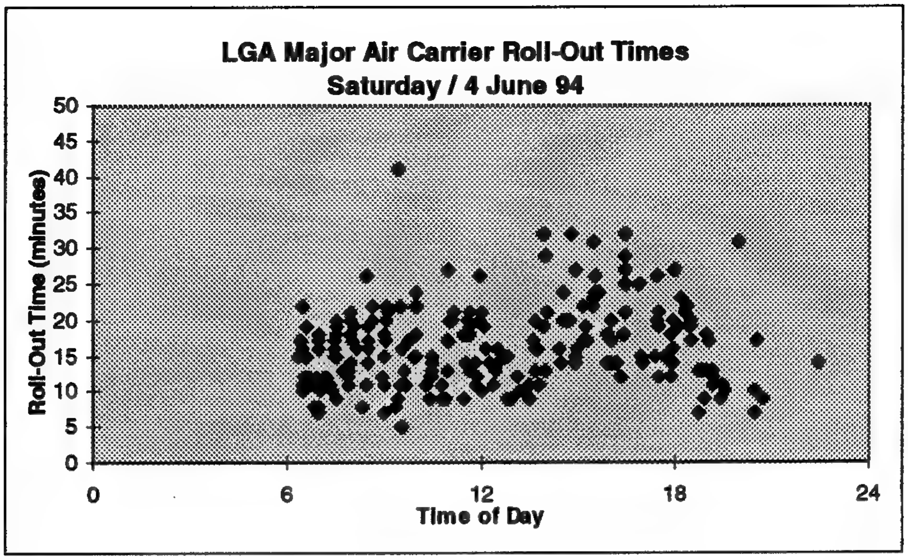


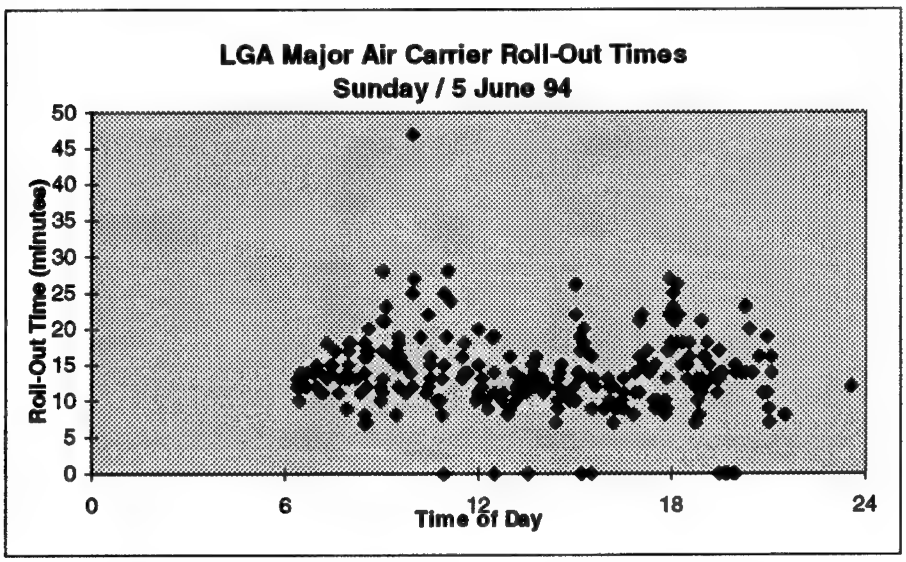



LGA Major Air Carrier Roll-Out Times
Wednesday / 1 June 94
Roll-Out Time (minutes)
50
45
40
35
30
25
20
15
10
5
0
0
6
12
18
24
Time of Day


LGA Major Air Carrier Roll-Out Times
Thursday / 2 June 94
Roll-Out Time (minutes)
50
45
40
35
30
25
20
15
10
5
0
0
6
12
18
24
Time of Day

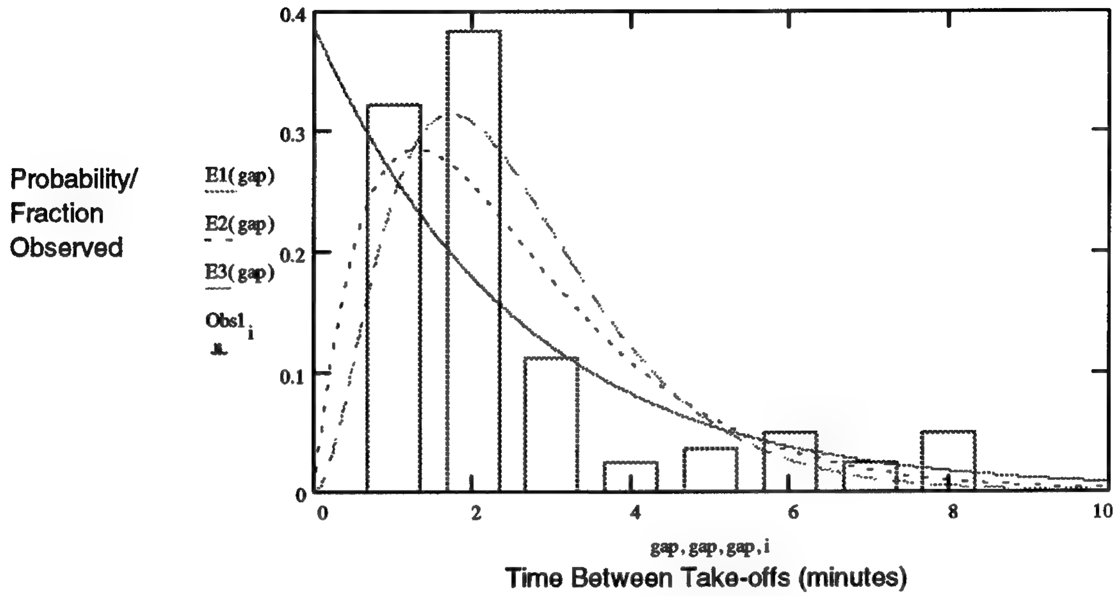

LGA Major Air Carrier Roll-out Times
Friday / 3 June 94
Roll-out Time (minutes)
50
45
40
35
30
25
20
15
10
5
0
0
6
12
18
24
Time of Day


LGA Major Air Carrier Roll-Out Times
Saturday / 4 June 94
Roll-Out Time (minutes)
0
5
10
15
20
25
30
35
40
45
50
0
6
12
18
24
Time of Day

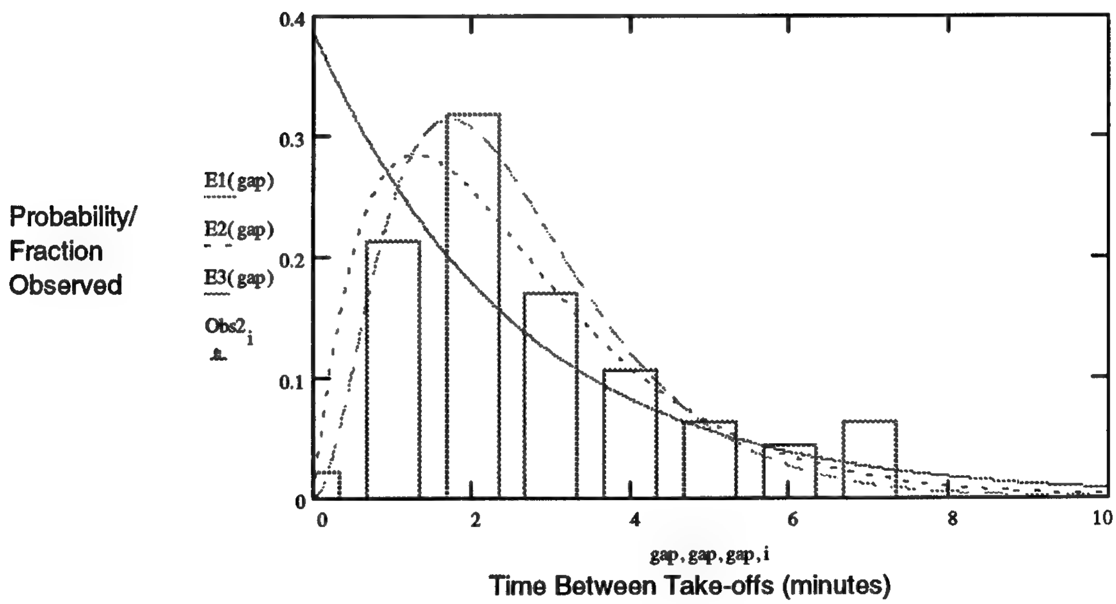

LGA Major Air Carrier Roll-Out Times
Sunday / 5 June 94
Roll-Out Time (minutes)
0
5
10
15
20
25
30
35
40
45
50
0
6
12
18
24
Time of Day

Service Time Plots
6 June Peak Periods

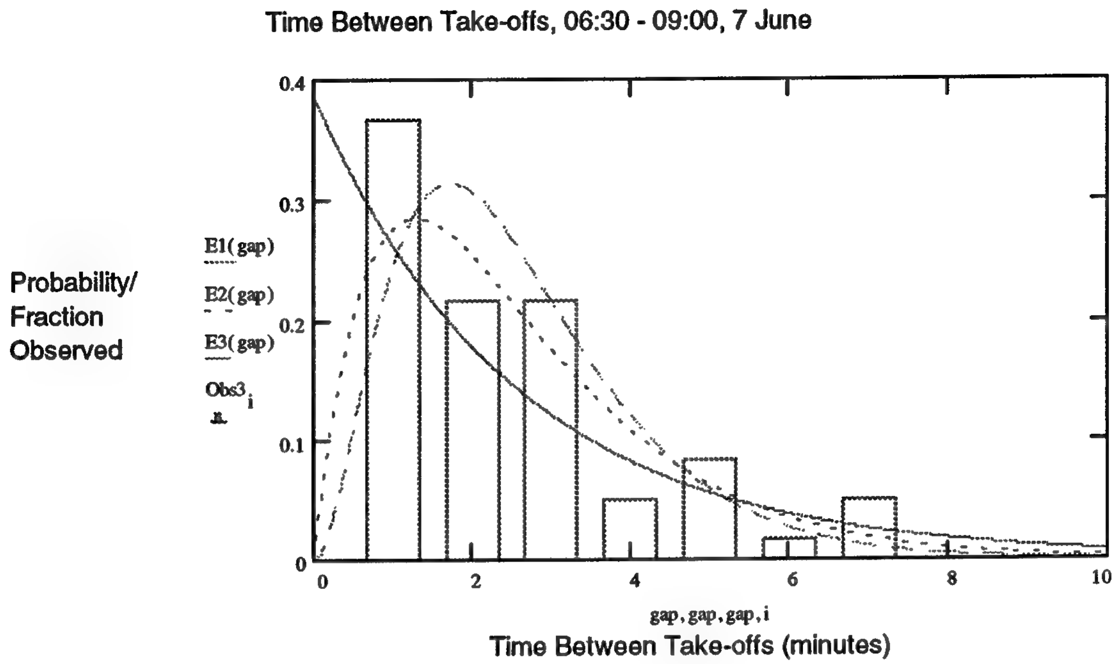
Time Between Take-offs, 06:30 - 10:00, 6 June

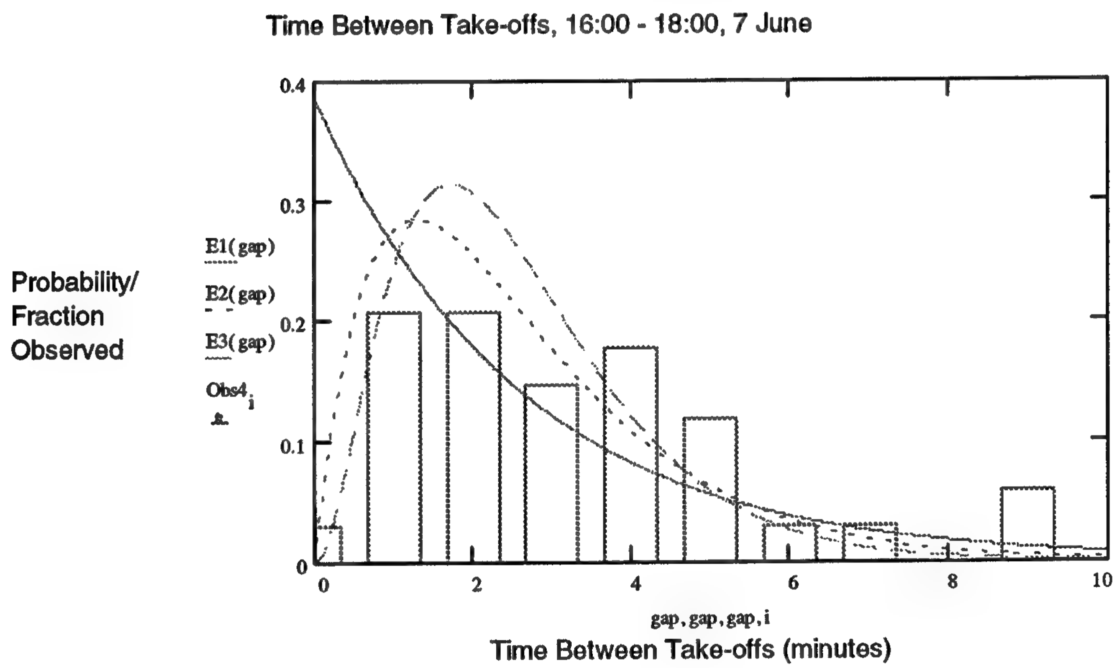


Time Between Take-offs, 15:00 - 18:00, 6 June



7 June Peak Periods





Appendix 2: Goodness of Fit Test Results

Kolmogorov-Smirnov Goodness of Fit test for the Exponential Dist.
6 June Hourly Gate Departure Times

decimal hour	elapsed minute	normalized times	i	D+	D-
6.3	0	0	0		
6.35	3	0.073171	1	-0.03	0.073
6.4	6	0.146341	2	-0.06	0.105
6.467	10	0.243902	3	-0.12	0.161
6.467	10	0.243902	4	-0.08	0.119
6.467	10	0.243902	5	-0.04	0.077
6.483	11	0.268293	6	-0.02	0.06
6.483	11	0.268293	7	0.023	0.018
6.5	12	0.292683	8	0.041	0.001
6.5	12	0.292683	9	0.082	-0.041
6.5	12	0.292683	10	0.124	-0.082
6.517	13	0.317073	11	0.141	-0.1
6.583	17	0.414634	12	0.085	-0.044
6.65	21	0.512195	13	0.029	0.012
6.65	21	0.512195	14	0.071	-0.029
6.667	22	0.536585	15	0.088	-0.047
6.817	31	0.756098	16	-0.09	0.131
6.867	34	0.829268	17	-0.12	0.163
6.883	35	0.853659	18	-0.1	0.145
6.95	39	0.95122	19	-0.16	0.201
6.983	41	1	20	-0.17	0.208
6.983	41	1	21	-0.13	0.167
6.983	41	1	22	-0.08	0.125
6.983	41	1	23	-0.04	0.083
6.983	41	1	24	-0	0.042

max D- max D-
0.141 0.208

Dcrit = 0.245, alpha = 0.1, 24 df

<u>**Conclusion**</u>: Do not reject fit at alpha = 0.1

decimal hour	elapsed minute	normalized times	i	D+	D-
7.0167	1	0.016667	0		
7.0167	1	0.016667	1	0.029	0.017
7.2333	14	0.233333	2	-0.142	0.188
7.2333	14	0.233333	3	-0.097	0.142
7.2667	16	0.266667	4	-0.085	0.13
7.2833	17	0.283333	5	-0.056	0.102
7.4	24	0.4	6	-0.127	0.173
7.4333	26	0.433333	7	-0.115	0.161
7.4667	28	0.466667	8	-0.103	0.148
7.5	30	0.5	9	-0.091	0.136
7.5	30	0.5	10	-0.045	0.091
7.5	30	0.5	11	0	0.045
7.5167	31	0.516667	12	0.029	0.017
7.5167	31	0.516667	13	0.074	-0.029
7.55	33	0.55	14	0.086	-0.041
7.55	33	0.55	15	0.132	-0.086
7.5667	34	0.566667	16	0.161	-0.115
7.5667	34	0.566667	17	0.206	-0.161
7.8833	53	0.883333	18	-0.065	0.111
7.9	54	0.9	19	-0.036	0.082
7.95	57	0.95	20	-0.041	0.086
7.95	57	0.95	21	0.005	0.041
7.9833	59	0.983333	22	0.017	0.029

max D+ max D-
0.206 0.188

Dcrit = 0.25, alpha = 0.1, 22 df

<u>**Conclusion**</u>: Do not reject fit at alpha = 0.1

decimal hour	elapsed minute	normalized times	i	D+	D-
8.017	1	0.016667	0		
8.05	3	0.05	1	-0	0.05
8.067	4	0.066667	2	0.029	0.019
8.183	11	0.183333	3	-0.04	0.088
8.233	14	0.233333	4	-0.04	0.09
8.417	25	0.416667	5	-0.18	0.226
8.467	28	0.466667	6	-0.18	0.229
8.483	29	0.483333	7	-0.15	0.198
8.483	29	0.483333	8	-0.1	0.15
8.483	29	0.483333	9	-0.05	0.102
8.5	30	0.5	10	-0.02	0.071
8.5	30	0.5	11	0.024	0.024
8.5	30	0.5	12	0.071	-0.024
8.533	32	0.533333	13	0.086	-0.038
8.55	33	0.55	14	0.117	-0.069
8.617	37	0.616667	15	0.098	-0.05
8.617	37	0.616667	16	0.145	-0.098
8.65	39	0.65	17	0.16	-0.112
8.7	42	0.7	18	0.157	-0.11
8.967	58	0.966667	19	-0.06	0.11
8.967	58	0.966667	20	-0.01	0.062
				max D-	max D-
				0.16	0.229

Dcrit = 0.26, alpha = 0.1, 21 df

Conclusion: Do not reject fit at alpha = 0.1

decimal hour	elapsed minute	normalized times	i	D+	D-
9	0	0	0		
9	0	0	1	0.045	0
9.0333	2	0.033333	2	0.058	-0.012
9.05	3	0.05	3	0.086	-0.041
9.1	6	0.1	4	0.082	-0.036
9.2167	13	0.216667	5	0.011	0.035
9.45	27	0.45	6	-0.177	0.223
9.4667	28	0.466667	7	-0.148	0.194
9.5	30	0.5	8	-0.136	0.182
9.5	30	0.5	9	-0.091	0.136
9.6333	38	0.633333	10	-0.179	0.224
9.6667	40	0.666667	11	-0.167	0.212
9.7167	43	0.716667	12	-0.171	0.217
9.7333	44	0.733333	13	-0.142	0.188
9.75	45	0.75	14	-0.114	0.159
9.7833	47	0.783333	15	-0.102	0.147
9.8167	49	0.816667	16	-0.089	0.135
9.8667	52	0.866667	17	-0.094	0.139
9.9	54	0.9	18	-0.082	0.127
9.95	57	0.95	19	-0.086	0.132
9.9667	58	0.966667	20	-0.058	0.103
9.9833	59	0.983333	21	-0.029	0.074
				max D+	max D-
				0.201	0.081

Dcrit = 0.25, alpha = 0.1, 22 df

Conclusion: Do not reject fit at alpha = 0.1

10.07	4	0.066667	0		
10.3	18	0.3	1	-0.23	0.3
10.42	25	0.416667	2	-0.27	0.345
10.45	27	0.45	3	-0.24	0.307
10.5	30	0.5	4	-0.21	0.286
10.5	30	0.5	5	-0.14	0.214
10.6	36	0.6	6	-0.17	0.243
10.6	36	0.6	7	-0.1	0.171
10.67	40	0.666667	8	-0.1	0.167
10.77	46	0.766667	9	-0.12	0.195
10.83	50	0.833333	10	-0.12	0.19
10.93	56	0.933333	11	-0.15	0.219
10.93	56	0.933333	12	-0.08	0.148
10.95	57	0.95	13	-0.02	0.093
				max D-	max D-
				-0.02	0.345

Dcrit = 0.349, alpha = 0.05, 14 df

Conclusion: Do not reject fit at alpha = 0.05

11.017	1	0.016667	0		
11.1	6	0.1	1	-0.041	0.1
11.133	8	0.133333	2	-0.016	0.075
11.183	11	0.183333	3	-0.007	0.066
11.2	12	0.2	4	0.035	0.024
11.233	14	0.233333	5	0.061	-0.002
11.5	30	0.5	6	-0.147	0.206
11.5	30	0.5	7	-0.088	0.147
11.567	34	0.566667	8	-0.096	0.155
11.6	36	0.6	9	-0.071	0.129
11.633	38	0.633333	10	-0.045	0.104
11.633	38	0.633333	11	0.014	0.045
11.85	51	0.85	12	-0.144	0.203
11.85	51	0.85	13	-0.085	0.144
11.917	55	0.916667	14	-0.093	0.152
11.933	56	0.933333	15	-0.051	0.11
11.95	57	0.95	16	-0.009	0.068
11.967	58	0.966667	17	0.033	0.025
				max D+	max D-
				0.061	0.206

Dcrit = 0.286, alpha = 0.1, 17 df

Conclusion: Do not reject fit at alpha = 0.1

decimal hour	elapsed minute	normalized times	i	D+	D-
12	0	0	0		
12.05	3	0.05	1	0.021	0.05
12.23	14	0.233333	2	-0.09	0.162
12.23	14	0.233333	3	-0.02	0.09
12.23	14	0.233333	4	0.052	0.019
12.47	28	0.466667	5	-0.11	0.181
12.5	30	0.5	6	-0.07	0.143
12.5	30	0.5	7	0	0.071
12.5	30	0.5	8	0.071	0
12.53	32	0.533333	9	0.11	-0.038
12.68	41	0.683333	10	0.031	0.04
12.82	49	0.816667	11	-0.03	0.102
12.85	51	0.85	12	0.007	0.064
12.95	57	0.95	13	-0.02	0.093
12.95	57	0.95	14	0.05	0.021
				max D-	max D-
				0.16	0.181

Dcrit = 0.314, alpha = 0.1, 14 df
Conclusion: Do not reject fit at alpha = 0.1

decimal hour	elapsed minute	normalized times	i	D+	D-
13.117	7	0.116667	0		
13.117	7	0.116667	1	-0.061	0.117
13.167	10	0.166667	2	-0.056	0.111
13.183	11	0.183333	3	-0.017	0.072
13.233	14	0.233333	4	-0.011	0.067
13.267	16	0.266667	5	0.011	0.044
13.3	18	0.3	6	0.033	0.022
13.383	23	0.383333	7	0.006	0.05
13.5	30	0.5	8	-0.056	0.111
13.5	30	0.5	9	0	0.056
13.633	38	0.633333	10	-0.078	0.133
13.7	42	0.7	11	-0.089	0.144
13.717	43	0.716667	12	-0.05	0.106
13.733	44	0.733333	13	-0.011	0.067
13.817	49	0.816667	14	-0.039	0.094
13.933	56	0.933333	15	-0.1	0.156
13.95	57	0.95	16	-0.061	0.117
13.967	58	0.966667	17	-0.022	0.078
13.967	58	0.966667	18	0.033	0.022
				max D+	max D-
				0.033	0.156

Dcrit = 0.286, alpha = 0.1, 17 df
Conclusion: Do not reject fit at alpha = 0.1

14	0	0	0		
14.05	3	0.05	1	0.017	0.05
14.2	12	0.2	2	-0.07	0.133
14.4	24	0.4	3	-0.2	0.267
14.47	28	0.466667	4	-0.2	0.267
14.5	30	0.5	5	-0.17	0.233
14.5	30	0.5	6	-0.1	0.167
14.58	35	0.583333	7	-0.12	0.183
14.6	36	0.6	8	-0.07	0.133
14.72	43	0.716667	9	-0.12	0.183
14.82	49	0.816667	10	-0.15	0.217
14.93	56	0.933333	11	-0.2	0.267
14.93	56	0.933333	12	-0.13	0.2
14.93	56	0.933333	13	-0.07	0.133
14.93	56	0.933333	14	-0	0.067
14.97	58	0.966667	15	0.033	0.033
				max D-	max D-
				0.033	0.267

Dcrit = 0.304, alpha = 0.1, 15 df
Conclusion: Do not reject fit at alpha = 0.1

15.017	1	0.016667	0		
15.083	5	0.083333	1	-0.028	0.083
15.117	7	0.116667	2	-0.006	0.061
15.183	11	0.183333	3	-0.017	0.072
15.2	12	0.2	4	0.022	0.033
15.25	15	0.25	5	0.028	0.028
15.25	15	0.25	6	0.083	-0.028
15.283	17	0.283333	7	0.106	-0.05
15.283	17	0.283333	8	0.161	-0.106
15.367	22	0.366667	9	0.133	-0.078
15.5	30	0.5	10	0.056	0
15.5	30	0.5	11	0.111	-0.056
15.517	31	0.516667	12	0.15	-0.094
15.533	32	0.533333	13	0.189	-0.133
15.9	54	0.9	14	-0.122	0.178
15.933	56	0.933333	15	-0.1	0.156
15.967	58	0.966667	16	-0.078	0.133
15.983	59	0.983333	17	-0.039	0.094
15.983	59	0.983333	18	0.017	0.039
				max D+	max D-
				0.189	0.178

Dcrit = 0.278, alpha = 0.1, 18 df
Conclusion: Do not reject fit at alpha = 0.1

decimal hour	elapsed minute	normalized times	i	D+	D-
16.05	3	0.05	0		
16.17	10	0.166667	1	-0.1	0.167
16.22	13	0.216667	2	-0.09	0.154
16.28	17	0.283333	3	-0.1	0.158
16.35	21	0.35	4	-0.1	0.163
16.43	26	0.433333	5	-0.12	0.183
16.45	27	0.45	6	-0.07	0.137
16.45	27	0.45	7	-0.01	0.075
16.47	28	0.466667	8	0.033	0.029
16.5	30	0.5	9	0.063	0
16.5	30	0.5	10	0.125	-0.063
16.67	40	0.666667	11	0.021	0.042
16.92	55	0.916667	12	-0.17	0.229
16.93	56	0.933333	13	-0.12	0.183
16.95	57	0.95	14	-0.07	0.137
16.95	57	0.95	15	-0.01	0.075
16.98	59	0.983333	16	0.017	0.046
				max D-	max D-
				0.125	0.229

Dcrit = 0.295, alpha = 0.1, 16 df

Conclusion: Do not reject fit at alpha = 0.1

decimal hour	elapsed minute	normalized times	i	D+	D-
17	0	0	0		
17.033	2	0.033333	1	0.022	0.033
17.05	3	0.05	2	0.061	-0.006
17.067	4	0.066667	3	0.1	-0.044
17.183	11	0.183333	4	0.039	0.017
17.35	21	0.35	5	-0.072	0.128
17.417	25	0.416667	6	-0.083	0.139
17.45	27	0.45	7	-0.061	0.117
17.5	30	0.5	8	-0.056	0.111
17.5	30	0.5	9	0	0.056
17.517	31	0.516667	10	0.039	0.017
17.6	36	0.6	11	0.011	0.044
17.683	41	0.683333	12	-0.017	0.072
17.767	46	0.766667	13	-0.044	0.1
17.767	46	0.766667	14	0.011	0.044
17.867	52	0.866667	15	-0.033	0.089
17.9	54	0.9	16	-0.011	0.067
17.917	55	0.916667	17	0.028	0.028
17.95	57	0.95	18	0.05	0.006
				max D+	max D-
				0.1	0.139

Dcrit = 0.278, alpha = 0.1, 18 df

Conclusion: Do not reject fit at alpha = 0.1

18	0	0	0		
18.02	1	0.016667	1	0.029	0.017
18.12	7	0.116667	2	-0.03	0.071
18.15	9	0.15	3	-0.01	0.059
18.22	13	0.216667	4	-0.03	0.08
18.25	15	0.25	5	-0.02	0.068
18.28	17	0.283333	6	-0.01	0.056
18.42	25	0.416667	7	-0.1	0.144
18.52	31	0.516667	8	-0.15	0.198
18.52	31	0.516667	9	-0.11	0.153
18.55	33	0.55	10	-0.1	0.141
18.68	41	0.683333	11	-0.18	0.229
18.7	42	0.7	12	-0.15	0.2
18.7	42	0.7	13	-0.11	0.155
18.75	45	0.75	14	-0.11	0.159
18.77	46	0.766667	15	-0.08	0.13
18.82	49	0.816667	16	-0.09	0.135
18.83	50	0.833333	17	-0.06	0.106
18.92	55	0.916667	18	-0.1	0.144
18.93	56	0.933333	19	-0.07	0.115
18.95	57	0.95	20	-0.04	0.086
18.98	59	0.983333	21	-0.03	0.074
18.98	59	0.983333	22	0.017	0.029
				max D-	max D-
				0.029	0.229

Dcrit = 0.25, alpha = 0.1, 22 df

Conclusion: Do not reject fit at alpha = 0.1

19.033	2	0.033333	0		
19.083	5	0.083333	1	1E-15	0.083
19.117	7	0.116667	2	0.05	0.033
19.133	8	0.133333	3	0.117	-0.033
19.267	16	0.266667	4	0.067	0.017
19.283	17	0.283333	5	0.133	-0.05
19.467	28	0.466667	6	0.033	0.05
19.517	31	0.516667	7	0.067	0.017
19.617	37	0.616667	8	0.05	0.033
19.717	43	0.716667	9	0.033	0.05
19.8	48	0.8	10	0.033	0.05
19.917	55	0.916667	11	-1E-15	0.083
19.983	59	0.983333	12	0.017	0.067
				max D+	max D-
				0.133	0.083

Dcrit = 0.338, alpha = 0.1, 12 df

Conclusion: Do not reject fit at alpha = 0.1

decimal hour	elapsed minute	normalized times	i	D+	D-
20.02	1	0.004167	0		
20.38	23	0.095833	1	-0.03	0.096
20.58	35	0.145833	2	-0.01	0.079
20.95	57	0.2375	3	-0.04	0.104
20.97	58	0.241667	4	0.025	0.042
21.07	64	0.266667	5	0.067	-6E-17
21.22	73	0.304167	6	0.096	-0.029
21.32	79	0.329167	7	0.138	-0.071
21.33	80	0.333333	8	0.2	-0.133
21.55	93	0.3875	9	0.213	-0.146
21.7	102	0.425	10	0.242	-0.175
22.73	164	0.683333	11	0.05	0.017
22.92	175	0.729167	12	0.071	-0.004
23.33	200	0.833333	13	0.033	0.033
23.57	214	0.891667	14	0.042	0.025
23.82	229	0.954167	15	0.046	0.021
				max D-	max D-
				0.242	0.104

Dcrit = 0.304, alpha = 0.1, 15 df

Conclusion: Do not reject fit at alpha = 0.1

Chi Square Goodness of Fit Test for the Exponential Distribution
6 June, 06:20 - 08:20, Gate Departure Times

decimal hour	gap time	decimal hour	gap time
6.3		7.2333	13
6.35	3	7.2333	0
6.4	3	7.2667	2
6.4667	4	7.2833	1
6.4667	0	7.4	7
6.4667	0	7.4333	2
6.4833	1	7.4667	2
6.4833	0	7.5	2
6.5	1	7.5	0
6.5	0	7.5	0
6.5	0	7.5167	1
6.5167	1	7.5167	0
6.5833	4	7.55	2
6.65	4	7.55	0
6.65	0	7.5667	1
6.6667	1	7.5667	0
6.8167	9	7.8833	19
6.8667	3	7.9	1
6.8833	1	7.95	3
6.95	4	7.95	0
6.9833	2	7.9833	2
6.9833	0	8.0167	2
6.9833	0	8.05	2
6.9833	0	8.0667	1
6.9833	0	8.1833	7
7.0167	2	8.2333	3
7.0167	0	8.4167	11

Bin	*Frequency*
0	18
1	9
2	10
3	5
4	4
5	1
6	0
7	0
8	2
9	1
10	0
11	0
12	1
13	0
14	1
15	0
16	0
17	0
18	0
19	0
20	1
Sum	53

Estimate parameters from the data

mean = 2.3962264
lambda = 0.4173228

Cell	Expected #	Observed #	(Oi - Ei)2 / Ei
0	18.083165	18	0.0003825
1	11.913338	9	0.7124398
2	7.8486045	10	0.589723
3	5.170725	5	0.0056369
(4-20)	9.9715939	11	0.1060632

Chi-Square Test Statistic: 1.4142454

Critical value 63.2
alpha = 0.1, 50 df

<u>**Conclusion**</u>: Do not reject the fit at alpha = 0.1

6 June, 11:00 - 14:00, Gate Departure Times

decimal hour	gap time	decimal hour	gap time
11.017		12.5	2
11.1	5	12.5	0
11.133	2	12.5	0
11.183	3	12.533	2
11.2	1	12.683	9
11.233	2	12.817	8
11.5	16	12.85	2
11.5	0	12.95	6
11.567	4	12.95	0
11.6	2	13.117	10
11.633	2	13.117	0
11.633	0	13.167	3
11.85	13	13.183	1
11.85	0	13.233	3
11.917	4	13.267	2
11.933	1	13.3	2
11.95	1	13.383	5
11.967	1	13.5	7
12	2	13.5	0
12.05	3	13.633	8
12.233	11	13.7	4
12.233	0	13.717	1
12.233	0	13.733	1
12.467	14	13.817	5
		13.933	7
		13.95	1
		13.967	1
		13.967	0

Bin	*Freq*
0	11
1	4
2	14
3	3
4	5
5	2
6	2
7	0
8	4
9	0
10	1
11	2
12	0
13	0
14	1
15	1
16	0
17	1
18	0
19	0
20	0

sum = 51

Estimate parameters from the data

mean = 3.4705882
lambda = 0.2881356

Cell	Expected #	Observed #	(Oi - Ei)2 / Ei
0	17.400782	11	2.3544924
1	11.463778	4	4.8594782
2	7.5524308	14	5.5043403
(3-20)	14.570911	22	3.7877776

Chi-Square Test Statistic: 16.506088

Critical value 63.2
alpha = 0.1, 50 df

<u>**Conclusion**</u>: Do not reject the fit at alpha = 0.1

6 June, 15:00 - 18:00, Gate Departure Times

decimal hour	gap time	decimal hour	gap time
15.017		16.5	2
15.083	4	16.5	0
15.117	2	16.667	10
15.183	4	16.917	15
15.2	1	16.933	1
15.25	3	16.95	1
15.25	0	16.95	0
15.283	2	16.983	2
15.283	0	17	1
15.367	5	17.033	2
15.5	8	17.05	1
15.5	0	17.067	1
15.517	1	17.183	7
15.533	1	17.35	10
15.9	22	17.417	4
15.933	2	17.45	2
15.967	2	17.5	3
15.983	1	17.5	0
15.983	0	17.517	1
16.05	4	17.6	5
16.167	7	17.683	5
16.217	3	17.767	5
16.283	4	17.767	0
16.35	4	17.867	6
16.433	5	17.9	2
16.45	1	17.917	1
16.45	0	17.95	2
16.467	1		

Bin	*Frequency*
0	9
1	11
2	9
3	4
4	6
5	5
6	2
7	1
8	3
9	0
10	0
11	2
12	0
13	0
14	0
15	1
16	0
17	0
18	0
19	0
20	0
Sum =	53

Estimate parameters from the data

mean = 3.2592593
lambda = 0.3068182

Cell	Expected #	Observed #	$\frac{(Oi - Ei)2}{Ei}$
0	14.267644	9	1.9448253
1	10.49791	11	0.0240138
2	7.7241977	9	0.2107237
3	5.6833439	4	0.498588
4	4.1817156	6	0.7906224
5	3.076841	5	1.2020578
(6-20)	8.4515583	9	0.0355897

Chi-Square Test Statistic: 3.4687731

Critical value 63.2
alpha = 0.1, 50 df

Conclusion: Do not reject the fit at alpha = 0.1

Kolmogorov-Smirnov Goodness of Fit test for the Exponential Dist.
6 June AM peak period, 0630 - 0900

T.O. (service) times

decimal hour	elapsed minute	normalized times	i	D+	D-
6.55	0	0	0		
6.5833	2	0.013793	1	0.00345	0.0138
6.6167	4	0.027586	2	0.0069	0.0103
6.65	6	0.041379	3	0.01034	0.0069
6.6833	8	0.055172	4	0.01379	0.0034
6.7333	11	0.075862	5	0.01034	0.0069
6.7667	13	0.089655	6	0.01379	0.0034
6.7833	14	0.096552	7	0.02414	-0.0069
6.8167	16	0.110345	8	0.02759	-0.0103
6.8333	17	0.117241	9	0.03793	-0.0207
6.85	18	0.124138	10	0.04828	-0.031
6.8833	20	0.137931	11	0.05172	-0.0345
6.9	21	0.144828	12	0.06207	-0.0448
6.9167	22	0.151724	13	0.07241	-0.0552
6.95	24	0.165517	14	0.07586	-0.0586
6.9833	26	0.17931	15	0.07931	-0.0621
7	27	0.186207	16	0.08966	-0.0724
7.0167	28	0.193103	17	0.1	-0.0828
7.05	30	0.206897	18	0.10345	-0.0862
7.1333	35	0.241379	19	0.08621	-0.069
7.2333	41	0.282759	20	0.06207	-0.0448
7.2833	44	0.303448	21	0.05862	-0.0414
7.3667	49	0.337931	22	0.04138	-0.0241
7.3833	50	0.344828	23	0.05172	-0.0345
7.4167	52	0.358621	24	0.05517	-0.0379
7.45	54	0.372414	25	0.05862	-0.0414
7.5833	62	0.427586	26	0.02069	-0.0034
7.6333	65	0.448276	27	0.01724	2E-16
7.6667	67	0.462069	28	0.02069	-0.0034
7.7333	71	0.489655	29	0.01034	0.0069
7.75	72	0.496552	30	0.02069	-0.0034
7.8	75	0.517241	31	0.01724	0
7.8167	76	0.524138	32	0.02759	-0.0103
7.85	78	0.537931	33	0.03103	-0.0138
7.8833	80	0.551724	34	0.03448	-0.0172
7.9167	82	0.565517	35	0.03793	-0.0207
7.9333	83	0.572414	36	0.04828	-0.031
7.9667	85	0.586207	37	0.05172	-0.0345
8	87	0.6	38	0.05517	-0.0379
8.0167	88	0.606897	39	0.06552	-0.0483
8.0333	89	0.613793	40	0.07586	-0.0586
8.0667	91	0.627586	41	0.07931	-0.0621
8.1167	94	0.648276	42	0.07586	-0.0586

Absence occurrences: T.O. gap > 3 min

i	elapsed minute	normalized time	D+	D-
1	35	0.241379	-0.1645	0.24138
2	41	0.282759	-0.1289	0.20584
3	49	0.337931	-0.1072	0.18408
4	62	0.427586	-0.1199	0.19682
5	65	0.448276	-0.0637	0.14058
6	71	0.489655	-0.0281	0.10504
7	94	0.648276	-0.1098	0.18674

8.2167	100	0.689655	43	0.05172	-0.0345
8.2333	101	0.696552	44	0.06207	-0.0448
8.2667	103	0.710345	45	0.06552	-0.0483
8.3167	106	0.731034	46	0.06207	-0.0448
8.3667	109	0.751724	47	0.05862	-0.0414
8.4	111	0.765517	48	0.06207	-0.0448
8.4333	113	0.77931	49	0.06552	-0.0483
8.4667	115	0.793103	50	0.06897	-0.0517
8.6	123	0.848276	51	0.03103	-0.0138
8.7	129	0.889655	52	0.0069	0.0103
8.8167	136	0.937931	53	-0.0241	0.0414
8.8333	137	0.944828	54	-0.0138	0.031
8.85	138	0.951724	55	-0.0034	0.0207
8.8833	140	0.965517	56	2.2E-16	0.0172
8.95	144	0.993103	57	-0.0103	0.0276
8.9667	145	1	58	-2E-16	0.0172
				max D+	max D-
				0.10345	0.0414

8	100	0.689655	-0.0743	0.15119
9	109	0.751724	-0.0594	0.13634
10	123	0.848276	-0.079	0.15597
11	129	0.889655	-0.0435	0.12042
12	136	0.937931	-0.0149	0.09178
13	144	0.993103	0.0069	0.07003
			max D+	max D-
			0.0069	0.24138

Dcrit = 1.22/sqrt(58) = 0.16019
alpha = 0.1, 58 degrees of freedom

Dcrit = 0.325
alpha = 0.1, 13 degrees of freedom

Conclusion: Do not reject an exponential fit

Concl: Do not reject an exp fit

Kolmogorov-Smirnov Goodness of Fit test for the Exponential Dist.
6 June PM peak period, 1500 - 1800

T.O. (service) times

decimal hour	elapsed minute	normalized time	i	D+	D-
15	0	0	0	0	
15.083	5	0.027933	1	-0.0083	0.0279
15.117	7	0.039106	2	0.00011	0.0195
15.167	10	0.055866	3	0.00296	0.0167
15.2	12	0.067039	4	0.01139	0.0082
15.267	16	0.089385	5	0.00865	0.011
15.35	21	0.117318	6	0.00033	0.0193
15.4	24	0.134078	7	0.00318	0.0164
15.467	28	0.156425	8	0.00044	0.0192
15.483	29	0.162011	9	0.01446	0.0051
15.5	30	0.167598	10	0.02848	-0.0089
15.533	32	0.178771	11	0.03692	-0.0173
15.55	33	0.184358	12	0.05094	-0.0313
15.55	33	0.184358	13	0.07054	-0.0509
15.65	39	0.217877	14	0.05663	-0.037
15.733	44	0.24581	15	0.04831	-0.0287
15.767	46	0.256983	16	0.05674	-0.0371
15.8	48	0.268156	17	0.06518	-0.0456
15.817	49	0.273743	18	0.0792	-0.0596
15.85	51	0.284916	19	0.08763	-0.068
16.217	73	0.407821	20	-0.0157	0.0353
16.333	80	0.446927	21	-0.0352	0.0548
16.367	82	0.458101	22	-0.0267	0.0463
16.383	83	0.463687	23	-0.0127	0.0323
16.4	84	0.469274	24	0.00131	0.0183
16.417	85	0.47486	25	0.01534	0.0043
16.533	92	0.513966	26	-0.0042	0.0238
16.55	93	0.519553	27	0.00986	0.0097
16.583	95	0.530726	28	0.01829	0.0013
16.667	100	0.558659	29	0.00997	0.0096
16.8	108	0.603352	30	-0.0151	0.0347
16.85	111	0.620112	31	-0.0123	0.0319
16.883	113	0.631285	32	-0.0038	0.0234
16.9	114	0.636872	33	0.01019	0.0094
16.933	116	0.648045	34	0.01862	0.001
17.017	121	0.675978	35	0.0103	0.0093
17.133	128	0.715084	36	-0.0092	0.0288
17.167	130	0.726257	37	-0.0008	0.0204
17.183	131	0.731844	38	0.01325	0.0064
17.217	133	0.743017	39	0.02169	-0.0021
17.433	146	0.815642	40	-0.0313	0.0509

Absence occurrences: T.O. gap > 3 min

i	elapsed minute	normalized time	D+	D-
1	12	0.067039	-0.0194	0.06704
2	16	0.089385	0.00585	0.04177
3	21	0.117318	0.02554	0.02208
4	24	0.134078	0.0564	-0.0088
5	33	0.184358	0.05374	-0.0061
6	39	0.217877	0.06784	-0.0202
7	51	0.284916	0.04842	-0.0008
8	73	0.407821	-0.0269	0.07449
9	85	0.47486	-0.0463	0.09391
10	95	0.530726	-0.0545	0.10215
11	100	0.558659	-0.0348	0.08247
12	108	0.603352	-0.0319	0.07954
13	116	0.648045	-0.029	0.07662
14	121	0.675978	-0.0093	0.05693
15	133	0.743017	-0.0287	0.07635
16	146	0.815642	-0.0537	0.10136

17.483	149	0.832402	41	-0.0285	0.0481
17.517	151	0.843575	42	-0.02	0.0397
17.567	154	0.860335	43	-0.0172	0.0368
17.583	155	0.865922	44	-0.0032	0.0228
17.617	157	0.877095	45	0.00526	0.0143
17.633	158	0.882682	46	0.01928	0.0003
17.667	160	0.893855	47	0.02771	-0.0081
17.75	165	0.921788	48	0.01939	0.0002
17.8	168	0.938547	49	0.02224	-0.0026
17.867	172	0.960894	50	0.0195	0.0001
17.983	179	1	51	-2E-16	0.0196
				0.08763	0.0548

17	151	0.843575	-0.0341	0.08167
18	160	0.893855	-0.0367	0.08433
19	165	0.921788	-0.017	0.06464
20	168	0.938547	0.01383	0.03379
21	172	0.960894	0.03911	0.00851
			0.06784	0.10215

Dcrit = 1.22/sqrt(51) = 0.17083
alpha = 0.1, 51 degrees of freedom

Dcrit = 0.325
alpha = 0.1, 13 degrees of freedom

Conclusion: Do not reject an exponential fit

Concl: Do not reject an exp. fit

Appendix 3: Justification for Numerical Solution Methods

A3.1 Uniformization

The matrix solution of the Kolmogorov Backward Equations can be determined through the use of uniformization. When v represents the uniformization rate and t is the length of the time period, the solution to the elements of the transition matrix, P(t), is:

$$P_{ij}(t) = \sum_{n=0}^{\infty} P_{ij}^{n} \cdot e^{-v \cdot t} \cdot \frac{(v \cdot t)^{n}}{n!} \qquad \text{(A3-1)}$$

The first part of the summation term, P_{ij}^{n}, represents the conditional probability of transitioning from state i to state j, given that n transitions are made. The second term represents the probability that n transitions occur during the time period. A major limitation of this form of the solution is the fact that many of the terms in the infinite sum must be evaluated in order to achieve a reasonable approximation (16:290). This process can turn out to be very computationally intensive. Another challenge is that the single step transition probabilities, P_{ij}, must be determined. The two approximation methods presented below address these problems. Both of these methods are from Sheldon Ross. (16:291)

A3.2 Approximation Method Number One

For the first approximation method, the value of the uniformization parameter v is selected so that v = n/t. By doing so, equation (A3-1) represents the expected value of the matrix product P^{N}, where N is Poisson distributed random variable with mean (vt). However, if N is Poisson with mean (vt), then it has a standard deviation equal to $(vt)^{1/2}$. When n is chosen to be large, then the mean of N (vt = n) is also large and the standard

deviation of N will be small in comparison to its mean. Thus, when n is large and equal to vt, the right side of equation (A3-1) is reasonably approximated by P^n.

It is now necessary to determine an estimate for P^n. The individual single step transition probabilities which are the elements of P are the following:

$$P_{ij} = \begin{cases} 1 - \dfrac{v_i}{v} & j = i \\ \dfrac{v_i}{v} \cdot P_{ij} & j \neq i \end{cases} \tag{A3-2}$$

Thus, when the rate matrix R is defined such that the elements R_{ij} ($i \neq j$) are equal to the transition rates from state i to state j, and the diagonal elements, R_{ii}, are equal to the negative of the total rate at which transitions occur out of state i, then the matrix with the individual elements P_{ij} is equal to:

$$P = I + \frac{R}{v} \tag{A3-3}$$

When v = n/t:

$$P = I + R \cdot \frac{t}{n} \tag{A3-4}$$

Thus, the n step transition probabilities are found in the n^{th} power of this matrix. Finally, the expected value of P^N is equal to the n^{th} power of (A3-4) as n goes to infinity.

$$P(t) = \lim_{n \to \infty} \left(I + R \cdot \frac{t}{n} \right)^n \tag{A3-5}$$

A3.3 Approximation Method Number Two

A random variable Y is defined to be exponentially distributed with rate λ. Then the conditional probability of transitioning to state j, at time Y, given the process started in state i, is expressed as follows:

$$P_{i,j} = P(X(Y) = j, \text{Given}, X(0) = i) \tag{A3-6}$$

Now the variable T_i is defined as the time at which the first transition occurs after time 0. By conditioning on where the random variable Y occurs with respect to T_i, the above transition probability may be rewritten as follows:

$$P_{i,j} = P\left(X(Y)=j, \text{Given}, X(0)=i, \text{and}, T_i \le Y\right) \cdot P\left(T_i \le Y\right) + P\left(X(Y)=j, \text{Given}, X(0)=i, \text{and}, T_i > Y\right) \cdot P\left(T_i > Y\right) \quad \text{(A3-7)}$$

But when T_i is > Y, the probability of transitioning from i to j by time Y is zero unless i is equal to j. Therefore, the following variable is defined to replace the second conditional probability expression in (A3-7) above:

$$\delta_i = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \quad \text{(A3-8)}$$

The probabilities that Y less than or equal to T_i, and that Y is greater than T_i are the following:

$$P\left(T_i \le Y\right) = \frac{v_i}{v_i + \lambda} \qquad P\left(T_i > Y\right) = \frac{\lambda}{v_i + \lambda} \quad \text{(A3-9)}$$

Also the following identity may be used to replace the first conditional probability expression in (A3-7):

$$P\left(X(Y)=j, \text{Given}, X(0)=i, \text{and}, T_i \le Y\right) = \sum_{(k \neq i)} P_{i,k} \cdot P_{k,j} \quad \text{(A3-10)}$$

Substituting (A3-8),(A3-9), and (A3-10) into equation (A3-7) yields:

$$P_{i,j} = \left[\sum_{(k \neq i)} P_{i,k} \cdot P_{k,j}\right] \cdot \left(\frac{v_i}{v_i + \lambda}\right) + \delta_i \cdot \left(\frac{\lambda}{v_i + \lambda}\right) \quad \text{(A3-11)}$$

When the variable i is equal to j, δ_i, is equal to 1. Also, $q_{ik} = P_{ik} v_i$. Thus (A3-11) becomes:

$$P_{i,j} = \left[\sum_{(k \neq i)} q_{i,k} \cdot P_{k,j}\right] \cdot \left(\frac{1}{v_i + \lambda}\right) + \left(\frac{\lambda}{v_i + \lambda}\right) \quad \text{(A3-12)}$$

$$\left(\lambda + v_i\right) \cdot P_{i,j} - \left[\sum_{(k \neq i)} q_{i,k} \cdot P_{k,j}\right] = \lambda \quad \text{(A3-13)}$$

$$\left(1+\frac{v_i}{\lambda}\right)\cdot P_{i,j}-\frac{1}{\lambda}\cdot\left[\sum_{(k\neq i)} q_{i,k}\cdot P_{k,j}\right]=1 \qquad \text{(A3-14)}$$

When i is not equal to j, then δ_i, is equal to zero and (A3-11) is:

$$P_{i,j}=\left[\sum_{(k\neq i)} q_{i,k}\cdot P_{k,j}\right]\cdot\frac{1}{v_i+\lambda} \qquad \text{(A3-15)}$$

$$(\lambda+v_i)\cdot P_{i,j}-\left[\sum_{(k\neq i)} q_{i,k}\cdot P_{k,j}\right]=0 \qquad \text{(A3-16)}$$

$$\left(1+\frac{v_i}{\lambda}\right)\cdot P_{i,j}-\frac{1}{\lambda}\cdot\left[\sum_{(k\neq i)} q_{i,k}\cdot P_{k,j}\right]=0 \qquad \text{(A3-17)}$$

Since the rate matrix R is defined such that:

$$R_{i,j} = \begin{cases} q_{i,j} & i\neq j \\ -v_i & i=j \end{cases} \qquad \text{(A3-18)}$$

It then follows that equations (A3-14) and (A3-17) are the equivalent expressions for the individual elements of the following matrix product:

$$\left(I-\frac{R}{\lambda}\right)\cdot P=I \qquad \text{(A3-19)}$$

Rearranging terms yields:

$$P=\left(I-\frac{R}{\lambda}\right)^{-1} \qquad \text{(A3-20)}$$

Finally, if the random variable Y is defined such that $Y = Y_1+Y_2...+Y_n$, where Y_i are all independent identically distributed exponential random variables, then:

$$P_{i,j} = P\left(X(Y_1+Y_2..Y_n)=j,\text{Given},X(0)=i\right) = \left(P^n\right)_{i,j} \qquad \text{(A3-21)}$$

If the parameter λ is equal to n/t in expression (A3-20), then the equation for the second approximation results:

$$P_{i,j} = \left[\left(I-\frac{R\cdot t}{n}\right)^{-1}\right]^n \qquad \text{(A3-22)}$$

Appendix 4: FORTRAN Code and User's Guide

This appendix contains the computer code for the four programs written to execute the models. The first two programs execute the Erlang model with the two different approximation methods. The second two programs execute the Absence model for the two methods.

A4.1 Erlang Models

The programs for the M/Ek/1 model are called "mek1.f" and "mek1inv.f". The first program uses approximation method one which does not require matrix inversions. The second program uses the matrix inversion approximation method. This program calls an IMSL double precision matrix inversion subroutine called "DLINRG" (8:1130). Both of these programs query the user for interactive inputs of model parameters. An example of this computer interaction is shown below.

```
Input an integer for the accuracy factor.
10
Input the size of the time step in hours.
1
Input the number of time periods.
6
Input the service rate (# of takeoffs per TIME PERIOD).
23
Input the number of stages of service.
2
Input an integer for the maximum # of aircraft in the system.
25
Input    6 periods of demand rate values.
25
23
23
20
14
19
Input the average time from pushback to queue entry.
10
Input the initial average delays in minutes (>= 0).
0
The initial state used has  0 aircraft in the system.
```

Figure A4.1 Erlang Model Computer Interaction

The accuracy factor is the number of matrix multiplications performed. If 6 is used, then 6 matrix multiplications are used to generate the 2^6 (32^{nd}) power of the matrix. Due to the characteristics of the rate matrix, it is possible for the "mek1.f" (no matrix inversion) program to abort if the accuracy factor is not chosen large enough. If this occurs, it is usually possible to perform a successful execution by increasing the magnitude of this factor. However, this should be done within reason, since the computation time increases linearly with the size of the accuracy factor.

The number of time periods is self explanatory.

The size of the time step should correspond to the interval for which the data was collected. If the interval was 30 minutes, the input would be 0.5.

The service rate is the greatest average number of take-offs that can be expected for the given airport operating conditions. This value should be input for the input time period length. The computer program will convert the input to an hourly rate, which is required for the rate matrix. For example, if it is expected that 10 take-offs may be performed in a 30 minute period, then the number 10 should be input for the service rate. The program will automatically convert this to an effective hourly rate of 20.

The number of stages of service should be determined by the closest fitting Erlang-k probability distribution. If the service time is assumed to be exponentially distributed, then the parameter would be 1. If the service time were actually more Gaussian in appearance, then a larger number of stages would be used.

The maximum number of aircraft in the system should be chosen to be 2 to 2.5 times larger than the longest expected queue length. This may be determined by executing

the model with an estimated value. If the value that was input did not meet this requirement, the model should be run again with a new value that does.

The next parameter requested is the number of aircraft which demand service per time period. These inputs are the number of aircraft which complete pushback per time period. If using the model for prediction, the number of OAG gate departures per time period will be used instead. Due to the requirements of the rate matrix, the computer programs will automatically convert the input values to the corresponding hourly rate.

Finally, the Erlang model program requests the initial average delay. The model will convert this amount to an appropriate queue length in order to determine the state probability vector for the model starting condition.

A4.2 Absences Models

The two programs which execute the Absence model are called "mekabs.f" and "mekabsinv.f". The first program does not require matrix inversions while the second does. As with the Erlang inversion model, "mekabsinv.f" calls an IMSL inversion subroutine. The requests for input are the same as those for the Erlang model programs up through the input of the demand rate values. An example of the additional computer interaction for the absence models is shown in Figure A4.2.

The next data input for the absence models after the demand rate values are the number of stages of service and absence return. These values must be input one per line. When using the absence model the service rate should have a lower variance and hence a higher number of stages of service than the Erlang models did. The absence return process will likely also have a small variance and thus require a large number of stages.

```
Input the number of stages of service and absence return.
6
9
Input    6 probabilities of an absence.
0.2
0.15
0.18
0.05
0.0
0.31
Input the absence return rate.
15
Input the initial number of aircraft in the system.
0
```

Figure A4.2 Absence Model Computer Interaction

The probability of an absence is the probability that the server (the runway) will not be available to allow another aircraft in the departure queue to take off after the present aircraft completes its take-off.

The absence return rate is the inverse of the average amount of time which the runway is unavailable during an absence occurrence. This must be computed using hours as the time unit. For example, if the average absence is 0.1 hours (6 minutes), then the absence return rate would be 1/(0.1) = 10.

Finally, these programs request the observed number of aircraft in the system as the starting condition. This amount is the number of aircraft in the queue, plus any aircraft taking off.

A4.3 Model Output

All programs generate three different output files. These files include the queue performance measure estimates, the matrix of state probability vectors, and the rate matrix.

Program	Queue Performance	Probability Vectors	Rate Matrix
mek1.f	perf.out	prob.out	rate.out
mek1inv.f	perfinv.out	probinv.out	rateinv.out
mekabs.f	perfabs.out	probabs.out	rateabs.out
mekabsinv.f	perfabsinv.out	probabsinv.out	rateabsinv.out

Table A3.1 Output File Names

The queue performance output file starts with an echo check of the important input values. It then lists the expected queue length and queue length standard deviation for the end of each time period. Next it lists the expected waiting time and the expected roll-out time. An example of a typical queue performance output file is shown below.

```
              M/Ek/1 MODEL with matrix inversions

 Time period length is:
    1.0
 Accuracy level factor is:
   10
 Maximum system size is:
   25
 Number of stages of service is:
    2
 The HOURLY service rate used was:
   23.0
 The average taxi-out time used was:
   10.0
 The HOURLY demand rates used were:
   25.0 23.0 23.0 20.0 14.0 19.0

 The queue length ave and standard deviation are:

    4.9   6.4   7.6   6.3   2.4   2.9
    4.1   5.5   6.2   6.1   4.0   3.8

 The ave delay and roll-out times are:

   14.5 18.5 21.6 18.3  7.8  9.1

   24.5 28.5 31.6 28.3 17.8 19.1
```

Figure A4.3 Sample Output File - "perfinv.out"

The following pages of this appendix contains the FORTRAN computer code.

The programs are listed in order: mek1.f, mek1inv.f, mekabs.f, and mekabsinv.f.

```
******************                 mek1.f           *******************
*  M(t)/Ek/1 Queue performance program, w/o inversions
*  Written by Joseph Hebert, Mar 95, Air Force Inst of Tech
*                     for masters thesis:
*  Analysis and Modeling of an Airport Departure Queue
********************************************************************
      PARAMETER (LDA=500,LDAINV=500,PER=100)
      DOUBLE PRECISION PMATRX(LDA,LDA), POUT(LDA,LDA)
      DOUBLE PRECISION SUM, PROD
      REAL IMATRX(LDA,LDA), PSTAGE(PER,LDA), RATE(LDA,LDA)
      REAL RSERVE, T, CHECK, IDELAY, NUM, TAXOUT
      REAL RARRIV(PER), NUMQ(PER), TIMQ(PER)
      REAL ENMQ2(PER), SDNUMQ(PER), ROLLTM(PER)
      INTEGER R, I, J, K, L, M, N, IHOUR, MAXNUM, ACC
      INTEGER  NSTAGE, NUMPER, SNUM

      DATA PSTAGE/50000*0.0/
      OPEN (UNIT=10,FILE='rate.out')
      OPEN (UNIT=11,FILE='prob.out')
      OPEN (UNIT=12,FILE='perf.out')

*  read in queue parameters

      PRINT*, 'Input an integer for the accuracy factor.'
      READ*,   ACC
      PRINT*, 'Input the size of the time step in hours.'
      READ*,   T
      PRINT*, 'Input the number of time periods.'
      READ*,   NUMPER
      PRINT*, 'Input the service rate (# of takeoffs per TIME PERIOD).'
      READ*,   RSERVE
      PRINT*, 'Input the number of stages of service.'
      READ*,   NSTAGE
      PRINT*, 'Input an integer for the maximum # of aircraft in the system'
      READ*,   MAXNUM
      PRINT*, 'Input ',NUMPER,' periods of demand rate values.'
      READ*,    (RARRIV(IHOUR), IHOUR = 1, NUMPER)
      PRINT*, 'Input the average time from pushback to queue entry.'
      READ*,   TAXOUT
      PRINT*, 'Input the initial average delays in minutes (>= 0).'
      READ*,   IDELAY

* define the matrix dimensions and the initial condition probability vector

      NUM = RSERVE*IDELAY/60
      SNUM = NINT(NUM)
      PRINT*,'The initial state used has',SNUM,' aircraft in the system.'
      N = 2**ACC
      R = MAXNUM*NSTAGE+1
      PSTAGE(1,NSTAGE*SNUM+1) = 1.0

* transform entry and service rates into hourly rates

      DO 5 I = 1, NUMPER
        RARRIV(I) = RARRIV(I)/T
5     CONTINUE
      RSERVE = RSERVE/T

*  create the r x r identity matrix

      DO 20 I = 1, R
        DO 10 J = 1, R
          IMATRX(I,J) = 0.0
10    CONTINUE
20    CONTINUE
      DO 30 I = 1, R
        IMATRX(I,I) = 1.0
30    CONTINUE
```

```
* compute rate matrices and perform numerical approximations

      DO 150 IHOUR = 1, NUMPER
        CALL RMATRX(IHOUR,RSERVE,RARRIV,NSTAGE,R,RATE,PER,LDA)
        DO 60 I = 1, R
          DO 50 J = 1, R
            PMATRX(I,J) = IMATRX(I,J)+(T/N)*RATE(I,J)
50        CONTINUE
60      CONTINUE

* check the rate matrix for errors

      DO 66 I = 1, R
        CHECK = 0.0
        DO 64 J = 1, R
          CHECK = CHECK + RATE(I,J)
64      CONTINUE
        IF (CHECK.GT.0.001.OR.CHECK.LT.-0.001) THEN
          PRINT*, 'There is an error in row ',I,' of the rate matrix'
          PRINT*, 'for hour number ',IHOUR,'.  Its row total is ',CHECK
          PRINT*, 'This program has been aborted.'
          GO TO 200
        ENDIF
66    CONTINUE

* matrix multiplication routine

      DO 120 M = 1, ACC
      DO 90 I = 1, R
        DO 80 J = 1, R
           PROD = 0.0
           SUM = 0.0
             DO 70 L = 1, R
               PROD = PMATRX(I,L)*PMATRX(L,J)
               SUM = PROD + SUM
70           CONTINUE
             POUT(I,J) = SUM
80       CONTINUE
90    CONTINUE

      DO 110 I = 1, R
        DO 100 J = 1, R
          PMATRX(I,J) = POUT(I,J)
100     CONTINUE
110   CONTINUE
120   CONTINUE

* check the transition matrix for probabilistic consistency

      DO 126 I = 1, R
        CHECK = 0.0
        DO 124 J = 1, R
          CHECK = CHECK + PMATRX(I,J)
124     CONTINUE
        IF (CHECK.GT.1.001.OR.CHECK.LT.0.999) THEN
          PRINT*, 'There is an error in row ',I,' of the trans matrix'
          PRINT*, 'for hour number ',IHOUR,'Its row total is ',CHECK
          PRINT*, 'This program has been aborted.'
          GO TO 200
        ENDIF
126   CONTINUE

* compute next probability vector

      DO 140 I = 1, R
        SUM = 0.0
        DO 130 J = 1, R
          PROD = PSTAGE(IHOUR,J)*PMATRX(J,I)
          SUM = PROD + SUM
```

```
130       CONTINUE
          PSTAGE(IHOUR+1,I) = SUM
140     CONTINUE
150     CONTINUE

* test printout of the prob matrix

        WRITE (11, '(A27)') ('mek1.f,The prob matrix is:')
        WRITE (11,'(A1)')
        DO 154 I = 1, NUMPER+1
          WRITE (11,'(1X,100F7.4)') (PSTAGE(I,J),J = 1, R)
          SUM = 0.0
          DO 153 J = 1, R
            SUM = SUM + PSTAGE(I,J)
153       CONTINUE
          WRITE (11,'(A,1F5.1)') 'Row sum: ',SUM
          WRITE (11,'(A1)')
154     CONTINUE

* calculate queue performance measures

        DO 180 I = 1, NUMPER
          NUMQ(I) = 0.0
          DO 170 J = 1, MAXNUM
            DO 160 K = 1, NSTAGE
              NUMQ(I) = NUMQ(I) + (J-1)*PSTAGE(I+1,(J-1)*NSTAGE+K+1)
              ENMQ2(I) = ENMQ2(I) + ((J-1)**2)*PSTAGE(I+1,(J-1)*NSTAGE+K+1)
160         CONTINUE
170       CONTINUE
          TIMQ(I) = 0.0
          DO 176 J = 1, R
            TIMQ(I) = TIMQ(I)+60*PSTAGE(I+1,J)*(J-1)/(NSTAGE*RSERVE)
176       CONTINUE
          SDNUMQ(I) = SQRT(ENMQ2(I) - NUMQ(I)**2)
          ROLLTM(I) = TAXOUT + TIMQ(I)
180     CONTINUE

* printout of the queue performance measures

        WRITE (12,'(A41)')('M/Ek/1 MODEL without inversions')
        WRITE (12,'(A60)')
        WRITE (12,'(A40)')('The time period length (hours) used was:')
        WRITE (12,'(1X,1F5.1)')(T)
        WRITE (12,'(A25)')('Accuracy level factor is:')
        WRITE (12,'(1X,1I3)')(ACC)
        WRITE (12,'(A23)')('Maximum system size is:')
        WRITE (12,'(1X,1I3)')(MAXNUM)
        WRITE (12,'(A31)')('Number of stages of service is:')
        WRITE (12,'(1X,1I3)')(NSTAGE)
        WRITE (12,'(A33)')('The HOURLY service rate used was:')
        WRITE (12,'(1X,1F5.1)')(RSERVE)
        WRITE (12,'(A35)')('The average taxi-out time used was:')
        WRITE (12,'(1X,1F5.1)')(TAXOUT)
        WRITE (12,'(A34)')('The HOURLY demand rates used were:')
        WRITE (12,'(1X,100F5.1)')(RARRIV(I), I = 1, NUMPER)
        WRITE (12,'(A60)')
        WRITE (12,'(A48)')('The queue length ave and standard deviation
       C are:')
        WRITE (12,'(A60)')
        WRITE (12,'(1X,100F5.1)')(NUMQ(I), I = 1, NUMPER)
        WRITE (12,'(1X,100F5.1)')(SDNUMQ(I), I = 1, NUMPER)
        WRITE (12,'(A60)')
        WRITE (12,'(A37)')('The ave delay and roll-out times are:')
        WRITE (12,'(A60)')
        WRITE (12,'(1X,100F5.1)')(TIMQ(I), I = 1, NUMPER)
        WRITE (12,'(A60)')
        WRITE (12,'(1X,100F5.1)')(ROLLTM(I), I = 1, NUMPER)

*test printout of the last rate matrix
```

```
200    WRITE (10,'(A32)') ('mek1.f, The last rate matrix is:')
       WRITE (10,'(A1)')
       DO 210 I = 1, R
         WRITE (10,'(1X,700F7.2)') (RATE(I,J),J = 1, R)
         WRITE (10,'(A1)')
210    CONTINUE

       END

* subroutine to create the rate matrix

       SUBROUTINE RMATRX(IHOUR,RSERVE,RARRIV,NSTAGE,R,RATE,PER,LDA)
       REAL RSERVE
       REAL RARRIV(PER), RATE(LDA,LDA)
       INTEGER R, NSTAGE, I, J, IHOUR

       DO 20 I = 1, R
         DO 10 J = 1, R
           RATE(I,J) = 0.0
10     CONTINUE
20     CONTINUE
       DO 30 I = 1, R-NSTAGE
         RATE(I,I) = -NSTAGE*RSERVE-RARRIV(IHOUR)
         RATE(I,I+NSTAGE) = RARRIV(IHOUR)
30     CONTINUE
       DO 40 I = 2, R
         RATE(I,I-1) = NSTAGE*RSERVE
40     CONTINUE
       RATE(1,1) = -RARRIV(IHOUR)
       DO 50 I = 1, NSTAGE
         RATE(R-I+1,R-I+1) = -NSTAGE*RSERVE
50     CONTINUE
       RETURN
       END
```

```
*********************          mek1inv.f          ******************
*  M(t)/Ek/1 Queue performance indicator program (with inversions)
*  Written by Joseph Hebert, Mar 95, Air Force Inst of Tech
*                   for masters thesis:
*  Analysis and Modeling of an Airport Departure Queue
**************************************************************
      PARAMETER (LDA=500,LDAINV=500,PER=100)
      DOUBLE PRECISION POUT(LDA,LDA),PMATRX(LDA,LDA)
      DOUBLE PRECISION SUM, PROD, PINV(LDAINV,LDAINV)
      REAL IMATRX(LDA,LDA), PSTAGE(PER,LDA), RATE(LDA,LDA)
      REAL RSERVE, T, CHECK, IDELAY, NUM, TAXOUT
      REAL RARRIV(PER), NUMQ(PER), TIMQ(PER)
      REAL ENMQ2(PER), SDNUMQ(PER), ROLLTM(PER)
      INTEGER R, I, J, K, L, M, N, IHOUR, MAXNUM, ACC
      INTEGER  NSTAGE, NUMPER, SNUM

      COMMON /WORKSP/  RWKSP
      REAL RWKSP(96740)
      CALL IWKIN(96740)

      DATA PSTAGE/50000*0.0/
      OPEN (UNIT = 10, FILE = 'rateinv.out')
      OPEN (UNIT = 11, FILE = 'probinv.out')
      OPEN (UNIT = 12, FILE = 'perfinv.out')

*  read in queue parameters

      PRINT*, 'Input an integer for the accuracy factor.'
      READ*,   ACC
      PRINT*, 'Input the size of the time step in hours.'
      READ*,   T
      PRINT*, 'Input the number of time periods.'
      READ*,   NUMPER
      PRINT*, 'Input the service rate (# of takeoffs per TIME PERIOD).'
      READ*,   RSERVE
      PRINT*, 'Input the number of stages of service.'
      READ*,   NSTAGE
      PRINT*, 'Input an integer for the maximum # of aircraft in the system'
      READ*,   MAXNUM
      PRINT*, 'Input ',NUMPER,' periods of demand rate values.'
      READ*,    (RARRIV(IHOUR), IHOUR = 1, NUMPER)
      PRINT*, 'Input the average time from pushback to queue entry.'
      READ*,   TAXOUT
      PRINT*, 'Input the initial average delays in minutes (>= 0).'
      READ*,   IDELAY

*define the matrix dimensions and the initial condition probability vector

      NUM = RSERVE*IDELAY/60
      SNUM = NINT(NUM)
      PRINT*,'The initial state used has',SNUM,' aircraft in the system.'
      N = 2**ACC
      R = MAXNUM*NSTAGE+1
      PSTAGE(1,NSTAGE*SNUM+1) = 1.0

* transform entry and service rates into hourly rates

      DO 5 I = 1, NUMPER
        RARRIV(I) = RARRIV(I)/T
5     CONTINUE
      RSERVE = RSERVE/T

*  create the r x r identity matrix

      DO 20 I = 1, R
        DO 10 J = 1, R
          IMATRX(I,J) = 0.0
10    CONTINUE
20    CONTINUE
```

```
      DO 30 I = 1, R
        IMATRX(I,I) = 1.0
30    CONTINUE

* compute rate matrices and perform numerical approximations

      DO 150 IHOUR = 1, NUMPER
        CALL RMATRX(IHOUR,RSERVE,RARRIV,NSTAGE,R,RATE,PER,LDA)
        DO 60 I = 1, R
          DO 50 J = 1, R
            PMATRX(I,J) = IMATRX(I,J)-(T/N)*RATE(I,J)
50        CONTINUE
60      CONTINUE

* check the rate matrix for errors

      DO 66 I = 1, R
        CHECK = 0.0
        DO 64 J = 1, R
          CHECK = CHECK + RATE(I,J)
64      CONTINUE
        IF (CHECK.GT.0.001.OR.CHECK.LT.-0.001) THEN
          PRINT*, 'There is an error in row ',I,' of the rate matrix'
          PRINT*, 'for hour number ',IHOUR,'.  Its row total is ',CHECK
          PRINT*, 'This program has been aborted.'
          GO TO 200
        ENDIF
66    CONTINUE

* call IMSL matrix inversion subroutine

      CALL DLINRG(R,PMATRX,LDA,PINV,LDAINV)

* matrix multiplication routine

      DO 120 M = 1, ACC
      DO 90 I = 1, R
        DO 80 J = 1, R
           PROD = 0.0
           SUM = 0.0
             DO 70 L = 1, R
               PROD = PINV(I,L)*PINV(L,J)
               SUM = PROD + SUM
70           CONTINUE
             POUT(I,J) = SUM
80       CONTINUE
90    CONTINUE

      DO 110 I = 1, R
        DO 100 J = 1, R
          PINV(I,J) = POUT(I,J)
100     CONTINUE
110   CONTINUE
120   CONTINUE

* check the transition matrix for probabilistic consistency

      DO 126 I = 1, R
        CHECK = 0.0
        DO 124 J = 1, R
          CHECK = CHECK + PINV(I,J)
124     CONTINUE
        IF (CHECK.GT.1.001.OR.CHECK.LT.0.999) THEN
          PRINT*, 'There is an error in row ',I,' of the trans matrix'
          PRINT*, 'for hour number ',IHOUR,'Its row total is ',CHECK
          PRINT*, 'This program has been aborted.'
          GO TO 200
        ENDIF
126   CONTINUE
```

```
* compute next probability vector

      DO 140 I = 1, R
        SUM = 0.0
        DO 130 J = 1, R
          PROD = PSTAGE(IHOUR,J)*PINV(J,I)
          SUM = PROD + SUM
130     CONTINUE
        PSTAGE(IHOUR+1,I) = SUM
140   CONTINUE
150   CONTINUE

* test printout of the prob matriX

      WRITE (11, '(A27)') ('mek1inv.f, The prob matrix is:')
      WRITE (11,'(A1)')
      DO 154 I = 1, NUMPER+1
        WRITE (11,'(1X,100F7.4)') (PSTAGE(I,J),J = 1, R)
        SUM = 0.0
        DO 153 J = 1, R
          SUM = SUM + PSTAGE(I,J)
153     CONTINUE
        WRITE (11,'(A,1F7.2)') 'Row sum: ',SUM
      WRITE (11,'(A1)')
154   CONTINUE

* calculate queue performance measures

      DO 180 I = 1, NUMPER
        NUMQ(I) = 0.0
        DO 170 J = 1, MAXNUM
          DO 160 K = 1, NSTAGE
            NUMQ(I) = NUMQ(I) + (J-1)*PSTAGE(I+1,(J-1)*NSTAGE+K+1)
            ENMQ2(I) = ENMQ2(I) + ((J-1)**2)*PSTAGE(I+1,(J-1)*NSTAGE+K+1)
160       CONTINUE
170     CONTINUE
        TIMQ(I) = 0.0
        DO 176 J = 1, R
          TIMQ(I) = TIMQ(I) + 60*PSTAGE(I+1,J)*(J-1)/(NSTAGE*RSERVE)
176     CONTINUE
        SDNUMQ(I) = SQRT(ENMQ2(I) - NUMQ(I)**2)
        ROLLTM(I) = TAXOUT + TIMQ(I)
180   CONTINUE

* printout of the queue performance measures

      WRITE (12,'(A45)')('M/Ek/1 MODEL with matrix inversions')
      WRITE (12,'(A60)')
      WRITE (12,'(A22)')('Time period length is:')
      WRITE (12,'(1X,1F5.1)')(T)
      WRITE (12,'(A25)')('Accuracy level factor is:')
      WRITE (12,'(1X,1I3)')(ACC)
      WRITE (12,'(A23)')('Maximum system size is:')
      WRITE (12,'(1X,1I3)')(MAXNUM)
      WRITE (12,'(A31)')('Number of stages of service is:')
      WRITE (12,'(1X,1I3)')(NSTAGE)
      WRITE (12,'(A33)')('The HOURLY service rate used was:')
      WRITE (12,'(1X,1F5.1)')(RSERVE)
      WRITE (12,'(A35)')('The average taxi-out time used was:')
      WRITE (12,'(1X,1F5.1)')(TAXOUT)
      WRITE (12,'(A34)')('The HOURLY demand rates used were:')
      WRITE (12,'(1X,100F5.1)')(RARRIV(I), I = 1, NUMPER)
      WRITE (12,'(A1)')
      WRITE (12,'(A48)')('The queue length ave and standard deviation
     C are:')
      WRITE (12,'(A60)')
      WRITE (12,'(1X,100F5.1)')(NUMQ(I), I = 1, NUMPER)
      WRITE (12,'(1X,100F5.1)')(SDNUMQ(I), I = 1, NUMPER)
```

```
      WRITE (12,'(A60)')
      WRITE (12,'(A37)')('The ave delay and roll-out times are:')
      WRITE (12,'(A60)')
      WRITE (12,'(1X,100F5.1)')(TIMQ(I), I = 1, NUMPER)
      WRITE (12,'(A60)')
      WRITE (12,'(1X,100F5.1)')(ROLLTM(I), I = 1, NUMPER)

*test printout of the last rate matrix

200   WRITE (10,'(A32)') ('mek1inv.f, The last rate matrix is:')
      WRITE (10,'(A1)')
      DO 210 I = 1, R
        WRITE (10,'(1X,700F7.2)') (RATE(I,J),J = 1, R)
        WRITE (10,'(A1)')
210   CONTINUE

      END

* subroutine to create the rate matrix

      SUBROUTINE RMATRX(IHOUR,RSERVE,RARRIV,NSTAGE,R,RATE,PER,LDA)
      REAL RSERVE
      REAL RARRIV(PER), RATE(LDA,LDA)
      INTEGER R, NSTAGE, I, J, IHOUR

      DO 20 I = 1, R
        DO 10 J = 1, R
          RATE(I,J) = 0.0
10    CONTINUE
20    CONTINUE
      DO 30 I = 1, R-NSTAGE
        RATE(I,I) = -NSTAGE*RSERVE-RARRIV(IHOUR)
        RATE(I,I+NSTAGE) = RARRIV(IHOUR)
30    CONTINUE
      DO 40 I = 2, R
        RATE(I,I-1) = NSTAGE*RSERVE
40    CONTINUE
      RATE(1,1) = -RARRIV(IHOUR)
      DO 50 I = 1, NSTAGE
        RATE(R-I+1,R-I+1) = -NSTAGE*RSERVE
50    CONTINUE
      RETURN
      END
```

```
*********************          mekabs.f          **********************
*  M(t)/Ek/1 ABSENCE Queue performance program, no matrix inversions
*  Written by Joseph Hebert, Mar 95, Air Force Inst of Tech
*                     for masters thesis:
*  Analysis and Modeling of an Airport Departure Queue
***********************************************************************
      PARAMETER (LDA=1000,PER=100)
      DOUBLE PRECISION PMATRX(LDA,LDA), POUT(LDA,LDA)
      DOUBLE PRECISION SUM, PROD
      REAL IMATRX(LDA,LDA), RATE(LDA,LDA), PSTAGE(PER,LDA)
      REAL RSERVE, PA(PER), RABSR, T, CHECK, TAXOUT
      REAL RARRIV(PER), NUMQ(PER), TIMQ(PER)
      REAL ENMQ2(PER), SDNUMQ(PER), ROLLTM(PER)
      INTEGER R, I, J, K, L, M, N, IHOUR, MAXNUM
      INTEGER  NS1, NS2, NUMPER, SNUM, ACC

      DATA PSTAGE/100000*0.0/
      OPEN (UNIT = 10, FILE = 'rateabs.out')
      OPEN (UNIT = 11, FILE = 'probabs.out')
      OPEN (UNIT = 12, FILE = 'perfabs.out')

*  read in queue parameters

      PRINT*, 'Input an integer for the accuracy factor.'
      READ*,   ACC
      PRINT*, 'Input the size of the time step in hours.'
      READ*,   T
      PRINT*, 'Input the number of time periods.'
      READ*,   NUMPER
      PRINT*, 'Input the service rate (takeoffs per TIME PERIOD).'
      READ*,   RSERVE
      PRINT*, 'Input an integer for the maximum # of aircraft in the system'
      READ*,   MAXNUM
      PRINT*, 'Input ',NUMPER,' periods of demand rate values.'
      READ*,   (RARRIV(IHOUR), IHOUR = 1, NUMPER)
      PRINT*, 'Input the number of stages of service and absence return.'
      READ*,   NS1, NS2
      PRINT*, 'Input ',NUMPER,' probabilities of an absence.'
      READ*,   (PA(IHOUR), IHOUR = 1, NUMPER)
      PRINT*, 'Input the absence return rate.'
      READ*,   RABSR
      PRINT*, 'Input the average time from pushback to queue entry.'
      READ*,   TAXOUT
      PRINT*, 'Input the initial number of aircraft in the system.'
      READ*,   SNUM

* define the matrix dimensions and the initial condition probability vector

      N = 2**ACC
      R = MAXNUM*(NS1+NS2)+NS2+1
      PSTAGE(1,(NS1+NS2)*SNUM+1) = 1.0

* transform entry and service rates into hourly rates

      DO 5 I = 1, NUMPER
        RARRIV(I) = RARRIV(I)/T
5     CONTINUE
      RSERVE = RSERVE/T

*  create the r x r identity matrix

      DO 20 I = 1, R
        DO 10 J = 1, R
          IMATRX(I,J) = 0.0
10    CONTINUE
20    CONTINUE
      DO 30 I = 1, R
        IMATRX(I,I) = 1.0
30    CONTINUE
```

```
* compute rate matrices and perform numerical approximations

      DO 180 IHOUR = 1, NUMPER
      CALL RMATRX(IHOUR,RSERVE,RARRIV,NS1,NS2,R,RATE,RABSR,PA,MAXNUM)
        DO 50 I = 1, R
          DO 40 J = 1, R
            PMATRX(I,J) = IMATRX(I,J)+(T/N)*RATE(I,J)
40        CONTINUE
50      CONTINUE

* check the rate matrix for errors

      DO 70 I = 1, R
        CHECK = 0.0
        DO 60 J = 1, R
          CHECK = CHECK + RATE(I,J)
60      CONTINUE
        IF (CHECK.GT.0.001.OR.CHECK.LT.-0.001) THEN
          PRINT*, 'There is an error in row ',I,' of the rate matrix'
          PRINT*, 'for hour number ',IHOUR,'.  Its row total is ',CHECK
          PRINT*, 'This program has been aborted.'
          GO TO 300
        ENDIF
70    CONTINUE

* matrix multiplication routine

      DO 130 M = 1, ACC
      DO 100 I = 1, R
        DO 90 J = 1, R
           PROD = 0.0
           SUM = 0.0
             DO 80 L = 1, R
               PROD = PMATRX(I,L)*PMATRX(L,J)
               SUM = PROD + SUM
80           CONTINUE
             POUT(I,J) = SUM
90       CONTINUE
100   CONTINUE

      DO 120 I = 1, R
        DO 110 J = 1, R
          PMATRX(I,J) = POUT(I,J)
110     CONTINUE
120   CONTINUE
130   CONTINUE

* check the transition matrix for probabilistic consistency

      DO 150 I = 1, R
        CHECK = 0.0
        DO 140 J = 1, R
          CHECK = CHECK + PMATRX(I,J)
140     CONTINUE
        IF (CHECK.GT.1.001.OR.CHECK.LT.0.999) THEN
          PRINT*, 'There is an error in row ',I,' of the trans matrix'
          PRINT*, 'for hour number ',IHOUR,'Its row total is ',CHECK
          PRINT*, 'This program has been aborted.'
          GO TO 300
        ENDIF
150   CONTINUE

* compute next probability vector

      DO 170 I = 1, R
        SUM = 0.0
        DO 160 J = 1, R
          PROD = PSTAGE(IHOUR,J)*PMATRX(J,I)
```

```
          SUM = PROD + SUM
160 -   CONTINUE
        PSTAGE(IHOUR+1,I) = SUM
170   CONTINUE
180   CONTINUE

* test printout of the prob matrix

      WRITE (11, '(A27)') ('mekabs.f, The prob matrix is:')
      WRITE (11,'(A1)')
      DO 200 I = 1, NUMPER+1
        WRITE (11,'(1X,700F7.4)') (PSTAGE(I,J),J = 1, R)
        SUM = 0.0
        DO 190 J = 1, R
          SUM = SUM + PSTAGE(I,J)
190     CONTINUE
        WRITE (11,'(A,1F8.4)') 'Row sum: ',SUM
        WRITE (11,'(A1)')
200   CONTINUE

* calculate queue performance measures

      DO 260 I = 1, NUMPER
       NUMQ(I) = 0.0
       DO 220 J = 1, MAXNUM
        DO 210 K = 1, (NS1+NS2)
       NUMQ(I) = NUMQ(I) + (J-1)*PSTAGE(I+1,J*(NS1+NS2)-NS1+K+1)
       ENMQ2(I) = ENMQ2(I) + ((J-1)**2)*PSTAGE(I+1,J*(NS1+NS2)-NS1+K+1)
210     CONTINUE
220    CONTINUE
       TIMQ(I) = 0.0
       DO 250 J = 1, MAXNUM
        DO 230 K = 1, NS1
          TIMQ(I) = TIMQ(I) + 60*PSTAGE(I+1,(NS1+NS2)*J+K-NS1+1)*
     C                 ((NS1*(J-1)+K)/(NS1*RSERVE)+((J-1)*PA(I))/RABSR)
230     CONTINUE
        DO 240 K = 1, NS2
         TIMQ(I) = TIMQ(I) + 60*PSTAGE(I+1,(NS1+NS2)*J+K+1)*
     C                 (J/RSERVE+K/(NS2*RABSR)+((J-1)*PA(I))/RABSR)
240     CONTINUE
250    CONTINUE
       SDNUMQ(I) = SQRT(ENMQ2(I) - NUMQ(I)**2)
       ROLLTM(I) = TAXOUT + TIMQ(I)
260   CONTINUE

* printout of the queue performance measures

      WRITE (12,'(A38)')('ABSENCE MODEL, mekabs.f')
      WRITE (12,'(A60)')
      WRITE (12,'(A40)')('The time period length (hours) used was:')
      WRITE (12,'(1X,1F5.1)')(T)
      WRITE (12,'(A25)')('Accuracy level factor is:')
      WRITE (12,'(1X,1I3)')(ACC)
      WRITE (12,'(A23)')('Maximum system size is:')
      WRITE (12,'(1X,1I3)')(MAXNUM)
      WRITE (12,'(A31)')('Number of stages of service is:')
      WRITE (12,'(1X,1I3)')(NS1)
      WRITE (12,'(A38)')('Number of stages of absence return is:')
      WRITE (12,'(1X,1I3)')(NS2)
      WRITE (12,'(A33)')('The HOURLY service rate used was:')
      WRITE (12,'(1X,1F5.1)')(RSERVE)
      WRITE (12,'(A35)')('The average taxi-out time used was:')
      WRITE (12,'(1X,1F5.1)')(TAXOUT)
      WRITE (12,'(A34)')('The HOURLY demand rates used were:')
      WRITE (12,'(1X,100F5.1)')(RARRIV(I), I = 1, NUMPER)
      WRITE (12,'(A43)')('The HOURLY absence probabilities used were:')
      WRITE (12,'(1X,100F6.3)')(PA(I), I = 1, NUMPER)
      WRITE (12,'(A60)')
      WRITE (12,'(A48)')('The queue length ave and standard deviation
```

```
     C are:')
      WRITE (12,'(A60)')
      WRITE (12,'(1X,100F5.1)')(NUMQ(I), I = 1, NUMPER)
      WRITE (12,'(1X,100F5.1)')(SDNUMQ(I), I = 1, NUMPER)
      WRITE (12,'(A60)')
      WRITE (12,'(A37)')('The ave delay and roll-out times are:')
      WRITE (12,'(A60)')
      WRITE (12,'(1X,100F5.1)')(TIMQ(I), I = 1, NUMPER)
      WRITE (12,'(A60)')
      WRITE (12,'(1X,100F5.1)')(ROLLTM(I), I = 1, NUMPER)

*test printout of the last rate matrix

300   WRITE (10,'(A32)') ('mekabs.f, The last rate matrix is:')
      WRITE (10,'(A1)')
      DO 310 I = 1, R
        WRITE (10,'(1X,700F9.4)') (RATE(I,J),J = 1, R)
        WRITE (10,'(A1)')
310   CONTINUE

      END

* subroutine to create the absence model rate matrix

      SUBROUTINE RMATRX(IHOUR,RSERVE,RARRIV,NS1,NS2,R,RATE,
     C     RABSR,PA,MAXNUM)
      REAL RSERVE, RABSR, PA(100)
      REAL RARRIV(100), RATE(1000,1000)
      INTEGER R, NS1, NS2, I, J, IHOUR, MAXNUM

      DO 20 I = 1, R
        DO 10 J = 1, R
          RATE(I,J) = 0.0
10    CONTINUE
20    CONTINUE
      DO 30 I = 1, R-NS1-NS2
        RATE(I,I+NS1+NS2) = RARRIV(IHOUR)
30    CONTINUE
      DO 40 I = 2, R
        RATE(I,I) = -NS2*RABSR-RARRIV(IHOUR)
        RATE(I,I-1) = NS2*RABSR
40    CONTINUE
      DO 60 I = 1, MAXNUM
        DO 50 J = 1, NS1
          RATE((NS1+NS2)*I-J+2,(NS1+NS2)*I-J+1) = NS1*RSERVE
                 RATE((NS1+NS2)*I-J+2,(NS1+NS2)*I-J+2) =
     C                          -RARRIV(IHOUR) - NS1*RSERVE
50      CONTINUE
        RATE((NS1+NS2)*I-NS1+2,(NS1+NS2)*I-NS1+1) =
     C                          PA(IHOUR)*NS1*RSERVE
        RATE((NS1+NS2)*I-NS1+2,(NS1+NS2)*(I-1)+1) =
     C                          (1-PA(IHOUR))*NS1*RSERVE
60    CONTINUE
      DO 70 I = 1, NS2
        RATE(R-I+1,R-I+1)=-NS2*RABSR
70    CONTINUE
      DO 80 I = 1, NS1
        RATE(R-NS2-I+1,R-NS2-I+1) = -NS1*RSERVE
80    CONTINUE
      RATE(1,1) = -RARRIV(IHOUR)
      RETURN
      END
```

```
******************         mekabsinv.f         *****************************
*  M(t)/Ek/1 ABSENCE Queue performance program, w/ matrix inversions
*  Written by Joseph Hebert, Mar 95, Air Force Inst of Tech
*                    for masters thesis:
*  Analysis and Modeling of an Airport Departure Queue
******************************************************************************
      PARAMETER (LDA=1000,LDAINV=1000,PER=100)
      DOUBLE PRECISION PMATRX(LDA,LDA), POUT(LDA,LDA)
      DOUBLE PRECISION SUM, PROD, PINV(LDAINV,LDAINV)
      REAL IMATRX(LDA,LDA), RATE(LDA,LDA), PSTAGE(PER,LDA)
      REAL RSERVE, PA(PER), RABSR, T, CHECK, TAXOUT
      REAL RARRIV(PER), NUMQ(PER), TIMQ(PER)
      REAL ENMQ2(PER), SDNUMQ(PER), ROLLTM(PER)
      INTEGER R, I, J, K, L, M, N, IHOUR, MAXNUM
      INTEGER  NS1, NS2, NUMPER, SNUM, ACC

      COMMON /WORKSP/  RWKSP
      REAL RWKSP(149016)
      CALL IWKIN(149016)

      DATA PSTAGE/100000*0.0/
      OPEN (UNIT = 10, FILE = 'rateabs.out')
      OPEN (UNIT = 11, FILE = 'probabs.out')
      OPEN (UNIT = 12, FILE = 'perfabsinv.out')

*  read in queue parameters

      PRINT*, 'Input an integer for the accuracy factor.'
      READ*,   ACC
      PRINT*, 'Input the size of the time step in hours.'
      READ*,   T
      PRINT*, 'Input the number of time periods.'
      READ*,   NUMPER
      PRINT*, 'Input the service rate (takeoffs per TIME PERIOD).'
      READ*,   RSERVE
      PRINT*, 'Input an integer for the maximum # of aircraft in the system'
      READ*,   MAXNUM
      PRINT*, 'Input ',NUMPER,' periods of demand rate values.'
      READ*,   (RARRIV(IHOUR), IHOUR = 1, NUMPER)
      PRINT*, 'Input the number of stages of service and absence return.'
      READ*,   NS1, NS2
      PRINT*, 'Input ',NUMPER,' probabilities of an absence.'
      READ*,   (PA(IHOUR), IHOUR = 1, NUMPER)
      PRINT*, 'Input the absence return rate.'
      READ*,   RABSR
      PRINT*, 'Input the average time from pushback to queue entry.'
      READ*,   TAXOUT
      PRINT*, 'Input the initial number of aircraft in the system.'
      READ*,   SNUM

* define the matrix dimensions and the initial condition probability vector

      N = 2**ACC
      R = MAXNUM*(NS1+NS2)+NS2+1
      PSTAGE(1,(NS1+NS2)*SNUM+1) = 1.0

* transform entry and service rates into hourly rates

      DO 5 I = 1, NUMPER
        RARRIV(I) = RARRIV(I)/T
5     CONTINUE
      RSERVE = RSERVE/T

*  create the r x r identity matrix

      DO 20 I = 1, R
        DO 10 J = 1, R
          IMATRX(I,J) = 0.0
10    CONTINUE
```

```
20     CONTINUE
       DO 30 I = 1, R
         IMATRX(I,I) = 1.0
30     CONTINUE

* compute rate matrices and perform numerical approximations

       DO 180 IHOUR = 1, NUMPER
       CALL RMATRX(IHOUR,RSERVE,RARRIV,NS1,NS2,R,RATE,RABSR,PA,MAXNUM)
         DO 50 I = 1, R
           DO 40 J = 1, R
             PMATRX(I,J) = IMATRX(I,J)-(T/N)*RATE(I,J)
40         CONTINUE
50       CONTINUE

* check the rate matrix for errors

       DO 70 I = 1, R
         CHECK = 0.0
         DO 60 J = 1, R
           CHECK = CHECK + RATE(I,J)
60       CONTINUE
         IF (CHECK.GT.0.001.OR.CHECK.LT.-0.001) THEN
           PRINT*, 'There is an error in row ',I,' of the rate matrix'
           PRINT*, 'for hour number ',IHOUR,'.  Its row total is ',CHECK
           PRINT*, 'This program has been aborted.'
           GO TO 300
         ENDIF
70     CONTINUE

* call IMSL double precision matrix inversion subroutine

       CALL DLINRG(R,PMATRX,LDA,PINV,LDAINV)

* matrix multiplication routine

       DO 130 M = 1, ACC
       DO 100 I = 1, R
         DO 90 J = 1, R
            PROD = 0.0
            SUM = 0.0
              DO 80 L = 1, R
                PROD = PINV(I,L)*PINV(L,J)
                SUM = PROD + SUM
80            CONTINUE
              POUT(I,J) = SUM
90        CONTINUE
100    CONTINUE

       DO 120 I = 1, R
         DO 110 J = 1, R
           PINV(I,J) = POUT(I,J)
110      CONTINUE
120    CONTINUE
130    CONTINUE

* check the transition matrix for probabilistic consistency

       DO 150 I = 1, R
         CHECK = 0.0
         DO 140 J = 1, R
           CHECK = CHECK + PINV(I,J)
140      CONTINUE
         IF (CHECK.GT.1.001.OR.CHECK.LT.0.999) THEN
           PRINT*, 'There is an error in row ',I,' of the trans matrix'
           PRINT*, 'for hour number ',IHOUR,'Its row total is ',CHECK
           PRINT*, 'This program has been aborted.'
           GO TO 300
         ENDIF
```

```
150    CONTINUE

* compute next probability vector

       DO 170 I = 1, R
         SUM = 0.0
         DO 160 J = 1, R
           PROD = PSTAGE(IHOUR,J)*PINV(J,I)
           SUM = PROD + SUM
160      CONTINUE
         PSTAGE(IHOUR+1,I) = SUM
170    CONTINUE
180    CONTINUE

* test printout of the prob matrix
       WRITE (11, '(A27)') ('mekabsinv.f, The prob matrix is:')
       WRITE (11,'(A1)')
       DO 200 I = 1, NUMPER+1
         WRITE (11,'(1X,700F7.4)') (PSTAGE(I,J),J = 1, R)
         SUM = 0.0
         DO 190 J = 1, R
           SUM = SUM + PSTAGE(I,J)
190      CONTINUE
         WRITE (11,'(A,1F8.4)') 'Row sum: ',SUM
         WRITE (11,'(A1)')
200    CONTINUE

* calculate queue performance measures
       DO 260 I = 1, NUMPER
        NUMQ(I) = 0.0
        DO 220 J = 1, MAXNUM
         DO 210 K = 1, (NS1+NS2)
        NUMQ(I) = NUMQ(I) + (J-1)*PSTAGE(I+1,J*(NS1+NS2)-NS1+K+1)
        ENMQ2(I) = ENMQ2(I) + ((J-1)**2)*PSTAGE(I+1,J*(NS1+NS2)-NS1+K+1)
210      CONTINUE
220     CONTINUE
        TIMQ(I) = 0.0
        DO 250 J = 1, MAXNUM
         DO 230 K = 1, NS1
           TIMQ(I) = TIMQ(I) + 60*PSTAGE(I+1,(NS1+NS2)*J+K-NS1+1)*
     C               ((NS1*(J-1)+K)/(NS1*RSERVE)+((J-1)*PA(I))/RABSR)
230      CONTINUE
         DO 240 K = 1, NS2
          TIMQ(I) = TIMQ(I) + 60*PSTAGE(I+1,(NS1+NS2)*J+K+1)*
     C             (J/RSERVE+K/(NS2*RABSR)+((J-1)*PA(I))/RABSR)
240      CONTINUE
250     CONTINUE
        SDNUMQ(I) = SQRT(ENMQ2(I) - NUMQ(I)**2)
        ROLLTM(I) = TAXOUT + TIMQ(I)
260    CONTINUE

* printout of the queue performance measures
       WRITE (12,'(A32)')('ABSENCE MODEL, mekabsinv.f')
       WRITE (12,'(A60)')
       WRITE (12,'(A40)')('The time period length (hours) used was:')
       WRITE (12,'(1X,1F5.1)')(T)
       WRITE (12,'(A25)')('Accuracy level factor is:')
       WRITE (12,'(1X,1I3)')(ACC)
       WRITE (12,'(A23)')('Maximum system size is:')
       WRITE (12,'(1X,1I3)')(MAXNUM)
       WRITE (12,'(A31)')('Number of stages of service is:')
       WRITE (12,'(1X,1I3)')(NS1)
       WRITE (12,'(A38)')('Number of stages of absence return is:')
       WRITE (12,'(1X,1I3)')(NS2)
       WRITE (12,'(A33)')('The HOURLY service rate used was:')
       WRITE (12,'(1X,1F5.1)')(RSERVE)
       WRITE (12,'(A35)')('The average taxi-out time used was:')
       WRITE (12,'(1X,1F5.1)')(TAXOUT)
       WRITE (12,'(A34)')('The HOURLY demand rates used were:')
```

```
      WRITE (12,'(1X,100F5.1)')(RARRIV(I), I = 1, NUMPER)
      WRITE (12,'(A43)')('The HOURLY absence probabilities used were:')
      WRITE (12,'(1X,100F6.3)')(PA(I), I = 1, NUMPER)
      WRITE (12,'(A60)')
      WRITE (12,'(A48)')('The queue length ave and standard deviation
     C are:')
      WRITE (12,'(A60)')
      WRITE (12,'(1X,100F5.1)')(NUMQ(I), I = 1, NUMPER)
      WRITE (12,'(1X,100F5.1)')(SDNUMQ(I), I = 1, NUMPER)
      WRITE (12,'(A60)')
      WRITE (12,'(A37)')('The ave delay and roll-out times are:')
      WRITE (12,'(A60)')
      WRITE (12,'(1X,100F5.1)')(TIMQ(I), I = 1, NUMPER)
      WRITE (12,'(A60)')
      WRITE (12,'(1X,100F5.1)')(ROLLTM(I), I = 1, NUMPER)

*test printout of the last rate matrix
300   WRITE (10,'(A32)') ('mekabsinv.f, The last rate matrix is:')
      WRITE (10,'(A1)')
      DO 310 I = 1, R
        WRITE (10,'(1X,700F9.4)') (RATE(I,J),J = 1, R)
        WRITE (10,'(A1)')
310   CONTINUE

      END

* subroutine to create the absence model rate matrix
      SUBROUTINE RMATRX(IHOUR,RSERVE,RARRIV,NS1,NS2,R,RATE,
     C      RABSR,PA,MAXNUM)
      REAL RSERVE, RABSR, PA(100)
      REAL RARRIV(100), RATE(1000,1000)
      INTEGER R, NS1, NS2, I, J, IHOUR, MAXNUM

      DO 20 I = 1, R
        DO 10 J = 1, R
          RATE(I,J) = 0.0
10    CONTINUE
20    CONTINUE
      DO 30 I = 1, R-NS1-NS2
        RATE(I,I+NS1+NS2) = RARRIV(IHOUR)
30    CONTINUE
      DO 40 I = 2, R
        RATE(I,I) = -NS2*RABSR-RARRIV(IHOUR)
        RATE(I,I-1) = NS2*RABSR
40    CONTINUE
      DO 60 I = 1, MAXNUM
        DO 50 J = 1, NS1
          RATE((NS1+NS2)*I-J+2,(NS1+NS2)*I-J+1) = NS1*RSERVE
                RATE((NS1+NS2)*I-J+2,(NS1+NS2)*I-J+2) =
     C                    -RARRIV(IHOUR) - NS1*RSERVE
50      CONTINUE
        RATE((NS1+NS2)*I-NS1+2,(NS1+NS2)*I-NS1+1) =
     C                    PA(IHOUR)*NS1*RSERVE
        RATE((NS1+NS2)*I-NS1+2,(NS1+NS2)*(I-1)+1) =
     C                    (1-PA(IHOUR))*NS1*RSERVE
60    CONTINUE
      DO 70 I = 1, NS2
        RATE(R-I+1,R-I+1)=-NS2*RABSR
70    CONTINUE
      DO 80 I = 1, NS1
        RATE(R-NS2-I+1,R-NS2-I+1) = -NS1*RSERVE
80    CONTINUE
      RATE(1,1) = -RARRIV(IHOUR)
      RETURN
      END
```

Appendix 5: Model Results

In order to clarify the observed roll-out times shown in Appendix 1, average roll-out times were computed for each hour of the day. These averages were calculated using the roll-out times for the period from 30 minutes before until 30 minutes after each hour. The average roll-out times are displayed in all the model output plots which follow.

A5.1 Friday, 3 June

The weather on 3 June was Visual Meteorological Conditions (VMC) throughout the day. The operating configuration was runway 22/31 from 6:00 to 7:00 AM, 4/31 from 7:00 AM to 1:00 PM, 31/4 from 1:00 to 4:00 PM, 22/31 from 4:00 to 5:00 PM, 22/13 from 5:00 to 9:00 PM, and 31/4 for the remainder of the day.

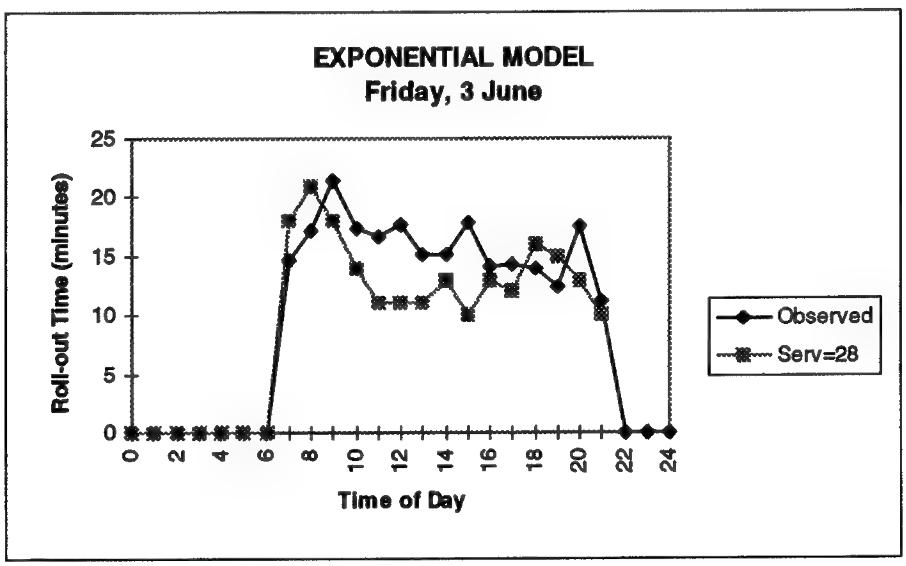


Figure A5.1 Exponential Model Results - 3 June

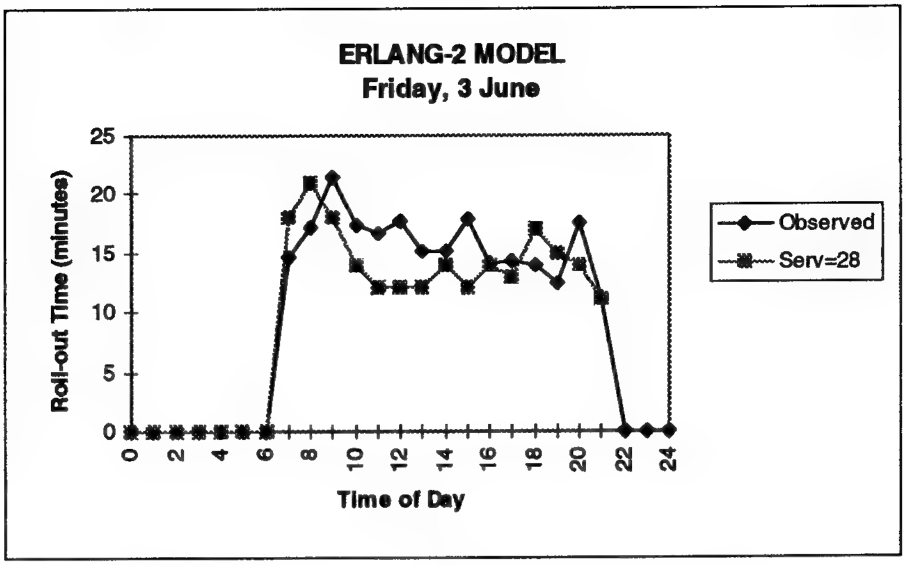


Figure A5.2 Erlang-2 Model Results - 3 June

A5.2 Saturday, 4 June

The departure demand on the weekends was observed to be significantly less than during the weekdays. The delays experienced were correspondingly lower. The weather on 4 June was VFR for the entire day. The operating runway configuration was 22/31 from 6:00 to 9:00 AM, 31/4 from 9:00 to 11:00 AM, 31/31 from 11:00 AM to 1:00 PM, and 13/13 for the rest of the day. The airport records indicate a runway closure occurred between 9:00 and 10:00 AM.

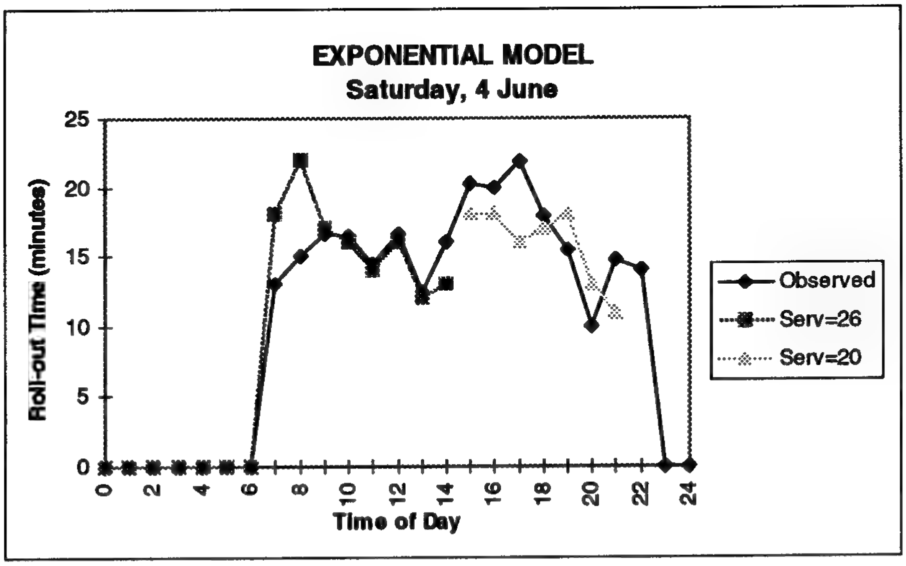


Figure A5.3 Exponential Model Results - 4 June

This output indicates that the airport had an effective service rate of 26 take-offs per hour for the first part of the day. The rate was significantly less during the afternoon.

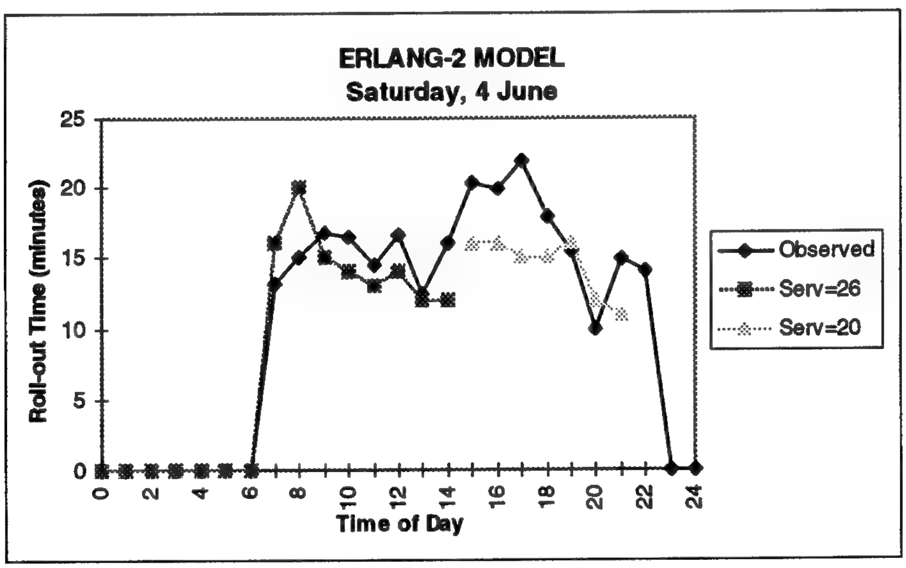


Figure A5.4 Erlang-2 Model Results - 4 June

This model result demonstrates that the effective service rate in the AM was roughly 26 take-offs per hour. The rate in the PM was significantly less. The results indicate that it was less than 21 take-offs per hour during this time.

A5.3 Sunday, 5 June

The weather was VFR for the entire day. The operating configuration was runway 13/13 from 6:00 to 11:00 AM, and 22/13 for the remainder of the day.

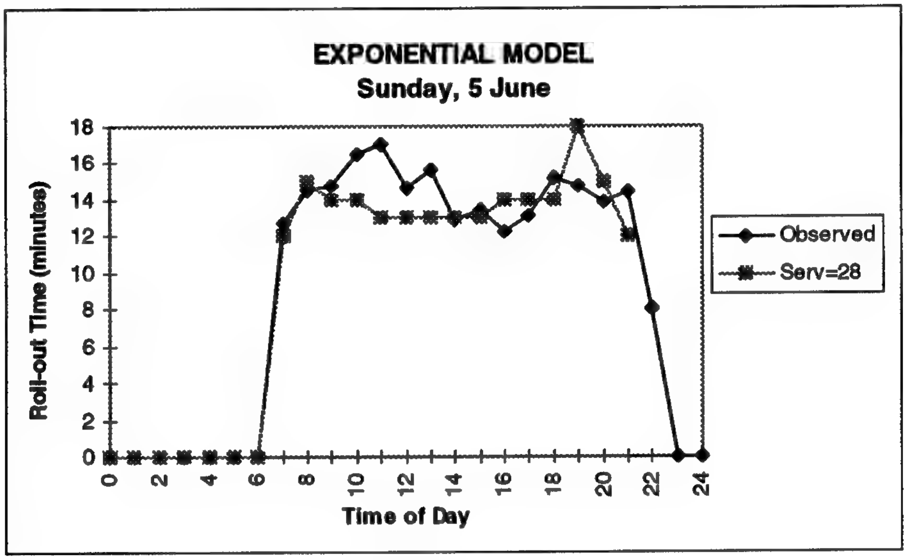


Figure A5.5 Exponential Model Results - 5 June

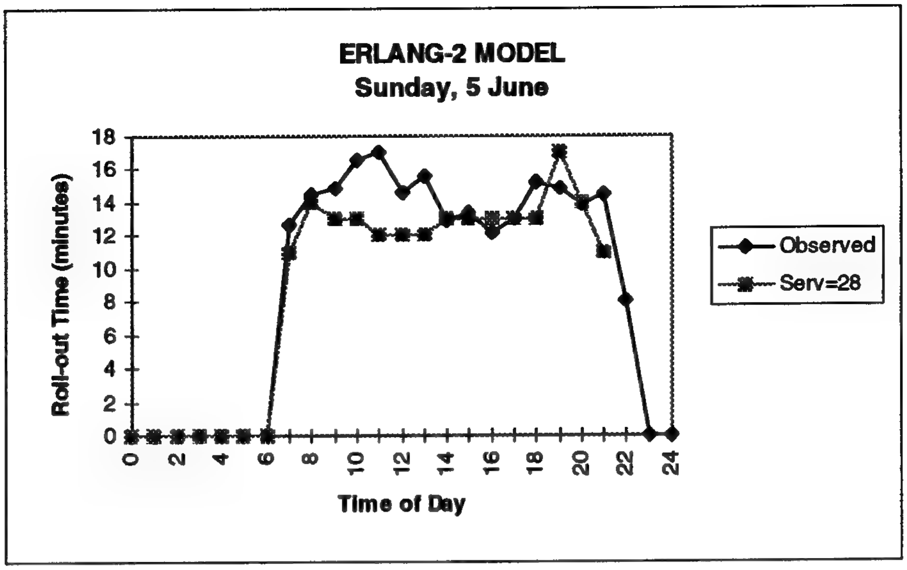


Figure A5.6 Erlang Model Results - 5 June

A5.4 Monday, 6 June

This day was one of the most interesting days in the study. There was a very significant pattern of delays experienced on this day. The weather was less than ideal for much of the day. The operating runway configuration was 22/13 throughout the day. The weather was reported as VFR1 from 6:00 to 9:00 AM, VFR2 from 9:00 to 11:00 AM, IFR from 11:00 AM to 1:00 PM, VFR2 from 1:00 PM to 7:00 PM, and VFR1 for the remainder of the day.

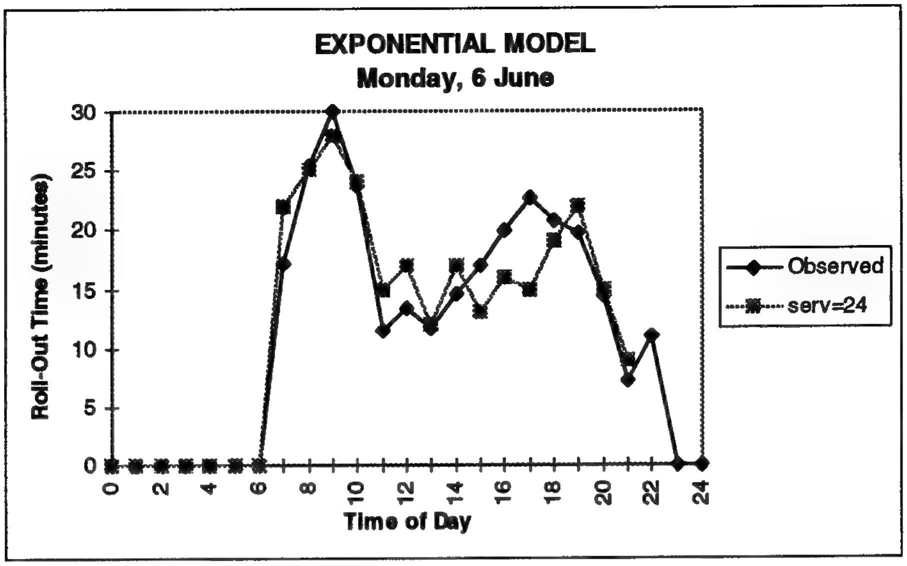


Figure A5.7 Exponential Model Results - 6 June

This model demonstrates a fairly good fit to the data when an effective service rate of 24 take-offs per hour is used.

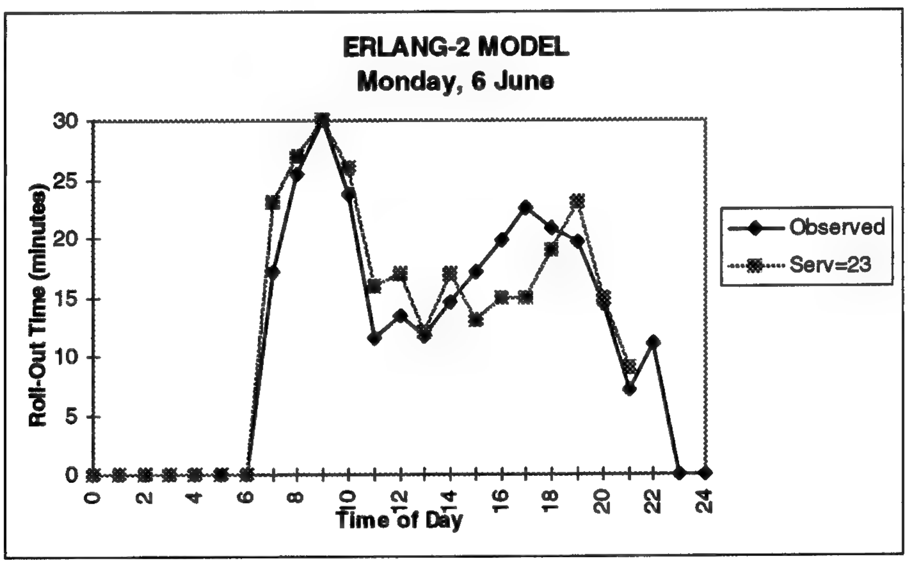


Figure A5.8 Erlang Model Results - 6 June

This graph demonstrates a similar ability to correlate variations in the roll-out time observed to the time-variant take-off demand process.

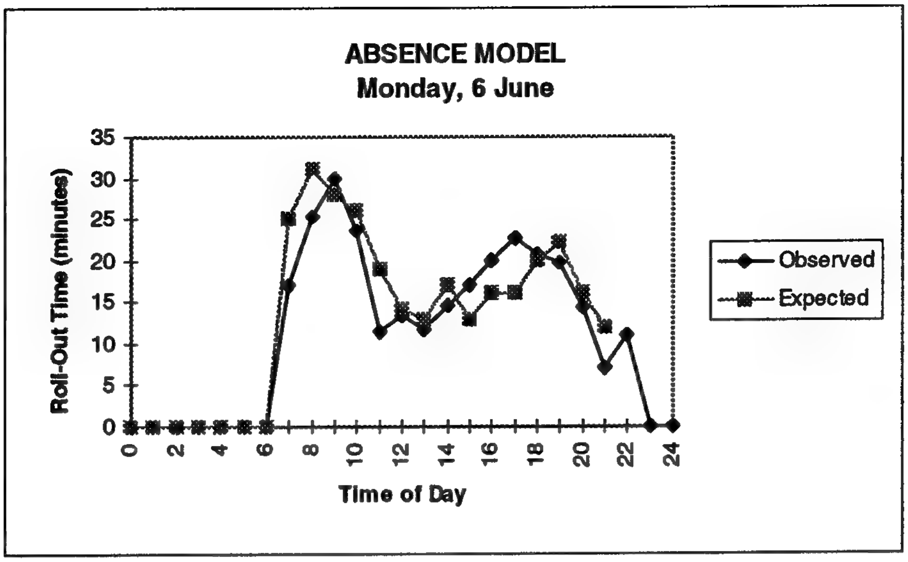


Figure A5.9 Absence Model Results - 6 June

This model also demonstrates a reasonable fit to the data. This model was performed assuming a take-off rate of 35 per hour. The probability that the runway would not be available to service an aircraft in this queue was estimated to be 0.2.

A5.5 Tuesday, 7 June

This day also experienced a significant weather pattern. In fact, it appeared to be worse than the weather experienced on the 6^{th} of June. In addition, the runway operating configurations used on this day had lower departure capacities. However, the resulting effective service rates for the models were higher. A possible explanation is given in Chapter 6, page 6-4. The weather was IFR from 06:00 to 8:00 AM, VFR2 from 8:00 to 9:00 AM, VFR1 from 9:00 to 10:00 AM, VFR2 from 10:00 to 11:00 AM, and VFR1 for the remainder of the day. The operating configuration was 22/13 from 6:00 to 7:00 AM, 22/31 from 7:00 AM to 12:00 PM, 31/4 from 12:00 to 1:00 PM, 22/31 from 1:00 to 4:00 PM , and 31/4 for the remainder of the day.

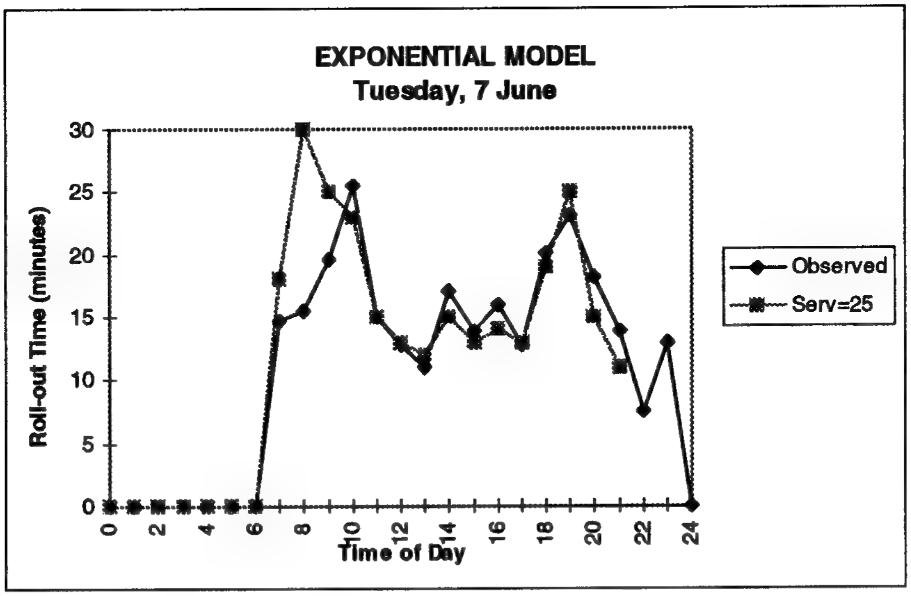


Figure A5.10 Exponential Model Results - 7 June

This model demonstrates good correlation to the roll-out times actually observed.

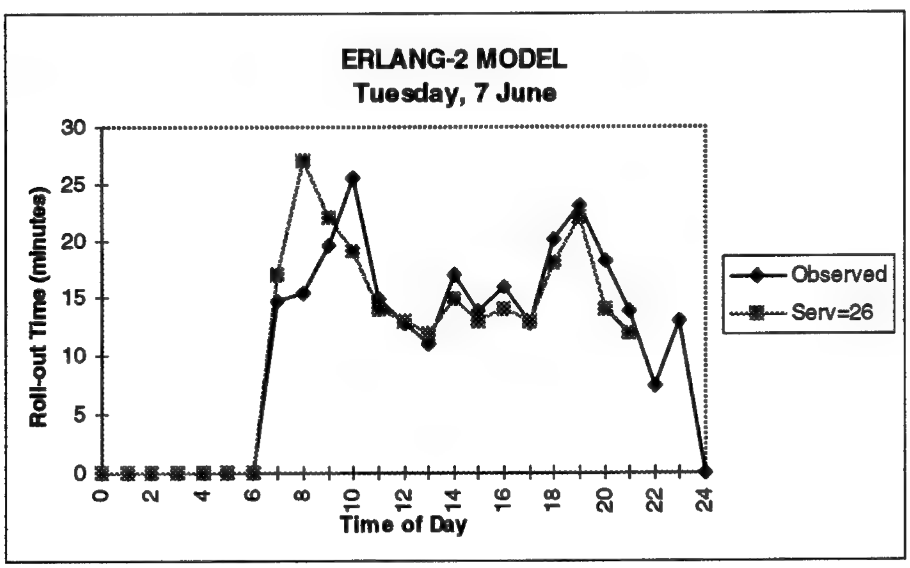


Figure A5.11 Erlang-2 Model Results - 7 June

The Erlang-2 model demonstrates a good fit for the data on the 7^{th} of June. It appears that the airport was operating with an effective departure rate of 26 take-offs per

hour. It also appears that the effective rate was somewhat greater for the first two hours of the day.

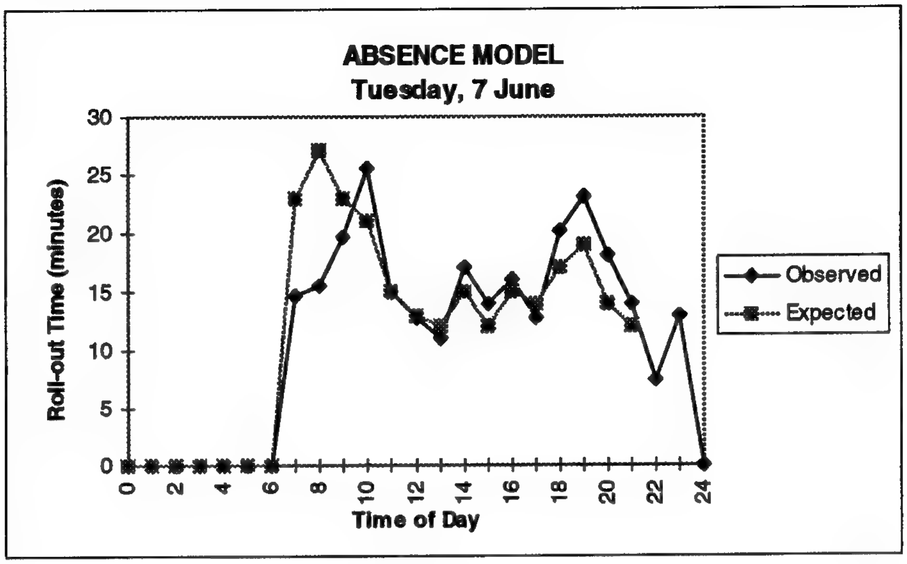


Figure A5.12 Absence Model Results - 7 June

These results were generated using a service rate of 35 take-offs per hour. By analyzing the peak AM period, the probability of an absence was estimated to be 0.23.

Bibliography

1. Baart, Douglas et al. Airport Capacity and Delay Analyses. Federal Aviation Administration (FAA) Report No. DOT/FAA/CT-TN91/18. Atlantic City International Airport, NJ: FAA Technical Center, 1991.

2. Banks, Jerry and John S. Carson II. Discrete Event System Simulation. Englewood Cliffs, NJ: Prentice Hall, Inc., 1984.

3. Department of Defense (DOD). DOD Flight Information Publication (Enroute), IFR - Supplement United States. MO: Defense Mapping Agency Aerospace Center, August - October 1994.

4. Federal Aviation Administration (FAA). LaGuardia Airport Data. FAA Smartflow Data Dictionary. Washington D. C.

5. Galliher, Herbert P. and R. Clyde Wheeler. "Nonstationary Queuing Probabilities for Landing Congestion of Aircraft," Operation Research Vol. 6, No. 3: 264-275 (January 1958).

6. Gilbo, Eugene P. "Airport Capacity: Representation, Estimation, Optimization." IEEE Transactions on Control Systems Technology, Vol. 1, No. 3: 144-154 (September 1993).

7. Gross, Donald and Douglas R. Miller. "The Randomization Technique as a Modeling Tool and Solution Procedure for Transient Markov Processes," Operations Research, Vol. 32, No. 2: 343-361 (March-April 1984).

8. IMSL. FORTRAN Subroutines for Mathematical Applications, Users Manual, IMSL Math/Library, TX, 1989.

9. Kleinrock, Leonard. Queueing Systems, Volume I: Theory. New York: John Wiley and Sons, Inc., 1975.

10. Koopman, Bernard O. "Air Terminal Queues under Time-Dependent Conditions," Operations Research, 20: 1089-1113 (Nov-Dec, 1972).

11. Krass, Dmitry et al. "A Weighted Markov Decision Process," Operations Research, 40: 1180-1189 (Nov-Dec 1992).

12. Medhi, J. Stochastic Models in Queueing Theory. San Diego: Academic Press, Inc., 1991.

13. Odoni, Amedeo R. et al. Flow Control of Congested Networks. Heidelberg, Germany: Springer-Verlag, 1987.

14. Richetta, Octavio and Amedeo R. Odoni. "Solving Optimally the Static Ground-Holding Delay Problem in Air Traffic Control," Transportation Science, 27: 228-238 (Aug 1993).

15. Ringel, Emanuel I. and Charles J Mode. "A Stochastic Analysis of State Transitions in an Air-Space Management System," Operations Research, 42: 262-273 (Mar-Apr 1994).

16. Ross, Sheldon M. Introduction to Probability Models (Fifth Edition). San Diego: Academic Press, Inc., 1993.

17. Ross, Sheldon M. Stochastic Processes. New York: John Wiley and Sons, Inc., 1983.

18. Roth, Emily. An Advanced Time-Dependent Queueing Model for Airport Delay Analysis. Report R79-9. Cambridge, MA: Massachusetts Institute of Technology Flight Transportation Laboratory, October 1979.

19. Rue, Robert C. and Matthew Rosenshine. Control of Entry to a Queueing System. Ph.D. thesis. The Pennsylvania State University, The Graduate School, University Park, PA, November 1979.

20. Rue, Robert C. and Matthew Rosenshine. "The Application of Semi-Markov Decision Processes to Queueing of Aircraft for Landing at an Airport," Transportation Science, 19: 154-172 (May 1985).

21. Shumsky, Robert. Dynamic Models for the Prediction of Take-Off Times: An Exploratory Data Analysis. Cambridge MA: Massachusetts Institute of Technology Operations Research Center, Sept 1993.

22. Vranas, Peter B. et al. "The Multi-Airport Ground-Holding Problem in Air Traffic Control," Operations Research, 42: 249-262 (Mar-Apr, 1994).

Vita

Major Joe Hebert was born on 16 July 1958 in Hartford, Connecticut. He graduated from South Windsor High School in South Windsor, Connecticut in June 1976. He then attended the United States Air Force Academy, graduating with a Bachelor of Science in Mathematics in May 1980. Upon graduation, he received a regular commission in the United States Air Force and attended Undergraduate Pilot Training at Williams Air Force Base, Arizona. His first flying assignment was to Spangdahlem Air Base, West Germany, as an Aircraft Commander in the F-4E and F-4G Wild Weasel. From there he was assigned as a T-37B Instructor Pilot at Reese Air Force Base, Texas. While at Reese he served as Flight Commander, Chief of Pilot Qualification Section, and Chief of Check Section. After Reese he was assigned to Moody Air Force Base, Georgia to fly the F-16A and F-16C. His duties there included Standardization and Evaluation Liaison Officer and Flight Commander. In February 1991, while stationed at Moody Air Force Base, he earned a Masters in Public Administration from Valdosta State College. Major Hebert was next assigned to Osan Air Base, South Korea, again flying the F-16C. His duties included Flight Commander and Assistant Operations Officer. In August 1993 he entered the Air Force Institute of Technology.

Permanent Address: 22 Raymond Rd
South Windsor, CT

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 95	Master's Thesis

4. TITLE AND SUBTITLE

ANALYSIS AND MODELING OF AN AIRPORT DEPARTURE PROCESS

5. FUNDING NUMBERS

6. AUTHOR(S)

Joseph E. Hebert, Major, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Department of Operational Sciences, ENS
Air Force Institute of Technology
WPAFB OH 45433-6583

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GOA/ENS/95M-02

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dan Citrenbaum (AOR-100)
Federal Aviation Administration
800 Independence Ave SW
Washington, DC 20591

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This study analyzes departure delays at a major airport using probability models to represent this non-homogeneous process. The models developed in this study expand on the Markovian models presently used by employing the method of stages to represent some of the model processes. This technique improves the user's ability to achieve a close fit for the service time probability distribution while maintaining the advantages of the Markovian model. The three types of models developed and compared all assume a Markovian system entry process. The first model uses an exponential distribution to model the service process. The second uses an Erlang distribution. The third models employs a unique server absence process to explicitly represents the periods of time when the server is unavailable to service departing aircraft. All three models generate results which correlate well with the delays actually observed. However, the Erlang model is preferred. Its results have lower variability than the exponential service time model. In addition, it generates solutions much faster than a typical application of the absence model. This model should be useful for improving capacity estimation and take-off delay prediction.

14. SUBJECT TERMS

Markov Processes, Stochastic Processes, Probability, Mathematical Models, Air Traffic, Uniformization, Randomization.

15. NUMBER OF PAGES

136

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to ***stay within the lines*** to meet ***optical scanning requirements***.

Block 1. Agency Use Only *(Leave blank)*.

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	- Contract	**PR**	- Project
G	- Grant	**TA**	- Task
PE	- Program Element	**WU**	- Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Name(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency Report Number. *(If known)*

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of...; To be published in.... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availability Statement. Denotes public availability or limitations. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR).

DOD - See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE - See authorities.
NASA - See Handbook NHB 2200.2.
NTIS - Leave blank.

Block 12b. Distribution Code.

DOD - Leave blank.
DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports.
NASA - Leave blank.
NTIS - Leave blank.

Block 13. Abstract. Include a brief *(Maximum 200 words)* factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code *(NTIS only)*.

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

★U.S.GPO:1990-0-273-271